An Interop Publication

ConneXions®

The Interoperability Report

October 1992 | *Special Issue: INTEROP 92 Fall Companion* | Volume 6, No. 10

ConneXions — The Interoperability Report tracks current and emerging standards and technologies within the computer and communications industry.

In this issue:

ConneXions is published monthly by Interop Company, 480 San Antonio Road, Suite 100, Mountain View, CA 94040, USA. 415-941-3399. Fax: 415-949-1779. Toll-free: 1-800-INTEROP.
E-mail: connexions@interop.com.

ISSN 0894–5926

From the Editor

Welcome to INTEROP 92 Fall and to San Francisco. I hope you will have some time during INTEROP week to see the city and the surrounding area. This issue of *ConneXions—The Interoperability Report* is designed to give you some "background reading" on a few of the conference and tutorial sessions. Of course, with so many sessions we can only give you a modest sampling, but *ConneXions* will continue to cover topics directly or indirectly related to what you'll hear this week.

This year, INTEROP has split into four sub-conferences to better address the needs of users across the entire spectrum of networking technologies. One such sub-conference is "SNA INTEROP," and our first article describes *Advanced Peer-to-Peer Networking* (APPN), a topic which should be of great interest to SNA users.

Network management continues to be a hot topic in both the user and standardization communities. In two articles we examine the OSF *Distributed Management Environment* (DME), and the *Simple Management Protocol* (SMP), the "next generation" SNMP.

As the world-wide Internet continues to grow at an alarming rate, researchers are looking at different ways to integrate the TCP/IP protocol suite with parts of the OSI suite. "Internet 2000" is a White Paper on one possible integration path.

Wireless networking coupled with powerful, low cost portable devices is likely to have a dramatic effect on the computer user of the 90s. *Personal Communications Services* (PCS) is one part of the wireless picture. A brief overview of PCS starts on page 32.

Many exciting new textbooks in the field of computer networking have been published this summer. Some of them will be available for the first time at INTEROP 92 Fall. In this issue we have reviews of no less than 5 books.

A debate about the viability of OSI is continuing in many places, including the "Letters to the Editor" pages of *ConneXions*. You are encouraged to voice your opinion on this and any other relevant topic. Send your letters to: *ConneXions,* 480 San Antonio Road, Mountain View, CA 94040, or via e-mail to: connexions@interop.com. We also welcome suggestions for topics you would like to see covered, as well as articles for publication. Keep in touch, and enjoy your visit to San Francisco!

Advanced Peer-to-Peer Networking (APPN): An Overview

by Steven T. Joyce and John Q. Walker II, IBM Corporation

Introduction

For the large base of *Advanced Program-to-Program Communication* (APPC) application users, one of their primary problems has been the amount of system configuration required. Using powerful *Advanced Peer-to-Peer Networking* (APPN) technology, computers can dynamically exchange almost all of the information that previously had to be configured by hand. APPN makes it simple to configure and maintain an SNA network.

APPC versus APPN

As their names imply, APPC, Advanced Program-to-Program Communication, deals with *programs,* while APPN, Advanced Peer-to-Peer Networking, deals with *networks.* (You may also hear APPC referred to as *LU 6.2.*) APPC defines the rules of how programs exchange information. These rules do not deal with the details of network setup and routing. It is APPN that defines how APPC traffic gets from one point to another in a network. A reasonable comparison between APPC and APPN is the difference between a person using the telephone and the services the telephone company offers.

• *APPC:* When a person wants to call someone, he looks up the telephone number and dials the telephone. Both parties identify themselves and the exchange of information begins. When the conversation is over, both parties say "good-bye" and hang up. This protocol, although informal, is generally accepted and makes it much easier to communicate. APPC provides the same functions and rules, only between application programs instead of people. An application program tells APPC with whom it needs a conversation. APPC starts a conversation between the programs so they can exchange data. When all the data has been exchanged, APPC provides a way for the programs to end the conversation.

• *APPN:* APPN provides networking functions similar to those provided by the telephone companies. After dialing a telephone number, the telephone network routes the call through trunks, switches, branches, and so on. To make the connection, the network takes into consideration what it knows about available routes and current problems. This happens without the caller understanding the details of the network. A person is able to talk on the telephone to another person no matter where they are or no matter how the call was routed. APPN provides these functions for APPC applications and their data. It computes routes for APPC communication through the network, dynamically calculating which route is best. Like the telephone company, APPN's routing is done transparently. APPC applications can't tell whether the communications partner in the APPN network is located in the same computer, one office away, or in another country. Similarly, if someone moves within the same city and takes their phone number, the phone network handles the change with no other user impact.

Network topology

Systems Network Architecture (SNA) has evolved from a heritage of mainframe computers, communications controllers, and terminals. With the increasing power of workstations and midrange computers, it has become more important to involve those computers in SNA networking. For many years, customers were required to configure networks in a hierarchical design. This hierarchical topology often lacks the flexibility to address varying network geographies, sizes, and workgroup relationships.

APPN meets modern flexibility requirements by allowing any network topology an enterprise wants to create. For example, each networked computer can be directly connected to every other computer (known as a "mesh") or they can all connect through a single routing "hub." Alternatively, some customers will choose to continue to use a hierarchical network design. Mesh, hub, and hierarchical networks, as well as mixtures of these, are all possible using APPN.

Key concepts

APPC is usually provided as system software. The APPC software provides two interfaces. The first, a programming interface "at the top," responds to requests from application programs that need to communicate. The second interface, "at the bottom," exchanges data with communications hardware.

A connection between the communications hardware on two computers is called a *link.* A link is generally started when the computers are powered on and their communications software is activated.

When an application wants to start communicating with another program, it issues an *Allocate call* to the APPC programming interface. The Allocate call includes the name of the destination, an *LU name* ("logical unit name"). The APPC software on each system is referred to as a *logical unit.* Thus, the LU name is a way to distinguish between different computers in the network. No two computers in an SNA network have the same LU name. This is similar to using Social Security numbers as a unique way to identify people. One difference is that a computer can have more than one LU at a time.

When an application issues an Allocate call, APPC sets up a *session* with the named partner LU. A session can be thought of as a pipe used to carry data between a pair of LUs. An LU can have more than one session with a partner LU and can talk to many different partner LUs at once. The new session uses the link already established between the communications hardware in the two computers.

To determine where partner LUs are located in the network, the computers in an APPN network, called *nodes,* exchange different types of messages. We will refer to these messages as *APPN control information.* At each node in an APPN network, one LU is selected to be the *control point LU.* The control point LU is used by APPN to exchange its control information. Normal APPC applications can also use the control point LU.

Dynamic configuration

APPN networks include three types of computers: low entry networking (LEN) nodes, end nodes (ENs), and network nodes (NNs).

LEN nodes, also known as "Type 2.1" nodes, have been available since the early 1980s. The LEN architecture was the first to allow computers in an SNA network to communicate with each other as peers. Examples of IBM platforms currently providing the LEN functions are APPC/PC for DOS, VTAM, and the RISC System/6000. (Note, however, that APPN end node and network node capabilities have been announced for VTAM and the RISC System/6000.) Many other software vendors ship LEN platforms, as well, such as Amdahl, Apple, DCA, DEC, Hewlett-Packard, Novell, Rabbit, Sun, and Unisys.

End nodes provide all of the functions of LEN nodes but also know how to use the services offered by APPN networks. For example, when end nodes connect to an APPN network they identify themselves, whereas LEN nodes don't. Also, when you start an APPC application, the end node works with the APPN network to find the application's partner. This makes setting up a network using end nodes easier than with LEN nodes.

continued on next page

Advanced Peer-to-Peer Networking *(continued)*

Network nodes provide all of the functions of end nodes and add two important services. First, network nodes work together to route information from one node to another. Network nodes providing this intermediate routing form the "backbone" of a network. The second service that network nodes provide is to help LEN and end nodes locate partner LUs in the network. By finding the LUs dynamically, very little system definition is required at each node in the network.

Example 1

The easiest way to learn how APPN works is with several examples of building and using a network. Figure 1 shows a simple APPN network. The lines that connect the computers are communications links.

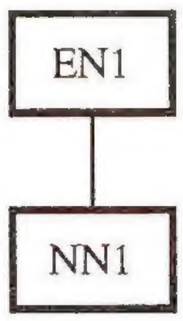


Figure 1: A simple APPN network, with one end node connected to a network node.

When the link is activated between end node 1 (EN1) and network node 1 (NN1) several things happen automatically:

- The computers tell each other that they are capable of supporting APPN, and the type of node they are: end node or network node.
- NN1 asks EN1 if it needs a *network node server*. Whenever an application on an end node needs to find an LU in the network, that end node sends its request to its network node server. Because EN1 does not have a network node server yet, it answers "yes." Although an end node can have connections to more than one node, it can only have one network node server at a time.
- Because NN1 will be serving EN1, they establish a pair of control point sessions. These are APPC sessions that will be used to exchange APPN control information. Two sessions are required, because the control point uses each session as a one-way pipe, combined to emulate full-duplex.
- EN1 registers any other APPC LUs that are defined at its node. It does this by sending NN1 a formatted record (that is, APPN control information) on the control point sessions.

Once these steps are complete, NN1 now knows how to get to EN1 and also knows what LUs are located there. This set of exchanges occurs every time a network node and an end node are joined by a communications link and agree to set up the EN–NN server relationship. The accumulation of this information by network nodes is crucial for locating LUs and calculating routes through the network.

Example 2

Different types of information are exchanged between a pair of network nodes.

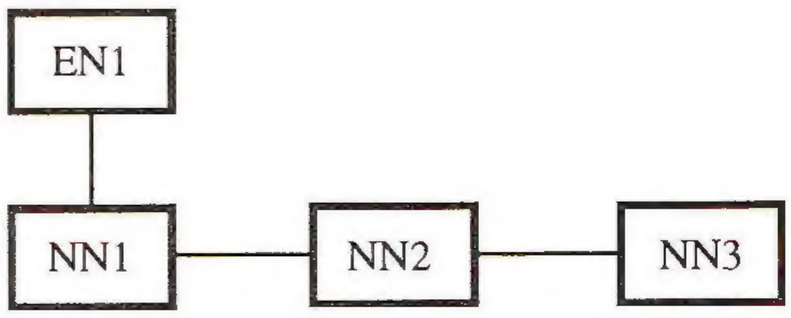


Figure 2: A sample APPN network with three network nodes and an end node.

Consider what happens when NN1 activates a link to NN2:

- The computers tell each other that they are capable of supporting APPN and are both network nodes.
- The network nodes bring up control point sessions between them to exchange APPN control information.
- Each network node in an APPN network keeps track of all the other network nodes and the links that connect them. This information is called the *network node topology*. Once NN1 and NN2 have control point sessions, they begin exchanging what they each know about the current network node topology. In this example, NN1 only knows about itself and its link to NN2. NN2 knows about NN3 and the link that joins them.
- Once they have exchanged topology information with one another, they both can construct a complete view of the layout of all the network nodes. However, NN3 must also have a complete network node topology. When a network node learns new topology information it spreads the word to any other network nodes it has links to. NN1 doesn't have any other network node links, so its job is complete. NN2 passes the new information it learned from NN1 on to NN3. NN3 doesn't have any other links to network nodes, so its job is also complete.

To summarize, once all the links are activated in the APPN network, each end node knows about itself (that is, its control point LU and any other LUs located at that computer) and its network node server. Each network node knows about itself, all the end nodes it serves, and the full network node topology—but not all the ENs and LUs in the network.

Before we leave this example, we should mention that network nodes can readily bridge between different types of links. For example, the link between end nodes and network nodes is frequently via a local area network (LAN). Links between network nodes are often wide area network (WAN) connections such as X.25 or SDLC. Network nodes provide this bridging for APPC traffic efficiently and transparently.

Locating resources

When LEN nodes connect directly to one another, they must have definitions for each partner LU with which they will exchange data. With today's growing and changing networks, keeping those definitions up-to-date can require a significant effort. In an APPN network, end nodes avoid requiring partner definitions by asking a network node to find the partner and the best route to get there. As described above, each end node tells its network node what LUs reside in it. Therefore, by combining the information known by all the network nodes in a network, the location of any LU can be determined. Let's take a look at a few examples:

Example 3

As seen in Figure 3, we will look at how LUs are found when both end nodes have the same network node server.

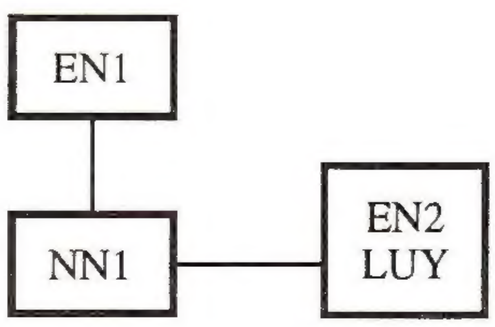


Figure 3: APPN network with two end nodes and one network node.

continued on next page

Advanced Peer-to-Peer Networking *(continued)*

When an APPC application on EN1 wants to start a conversation with an application on EN2:

- EN1 asks NN1 to find LUY and determine what path through the network should be used.
- NN1 knows that it is the network node server for EN2 and that EN2 has registered its LUs.
- NN1 determines that the only path available is "EN1 to NN1 to EN2." It passes this information back to EN1.
- An application in EN1 can now establish an APPC session to LUY and start exchanging information.

Example 4 Figure 4 shows an APPN network with additional complexity. Notice that EN3 has links to two network nodes. Assume that NN3 is the network node server for EN3.

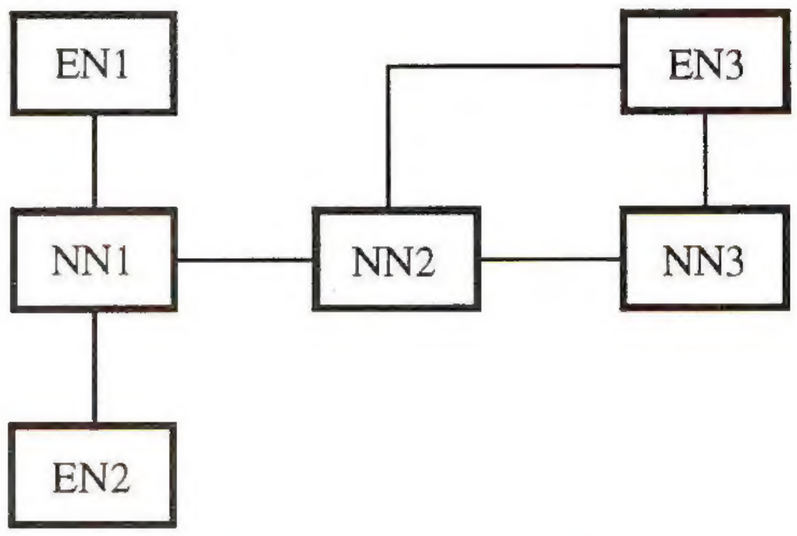


Figure 4: An APPN network with three network nodes and three end nodes.

When an APPC application on EN1 wants to start a conversation with an application on EN3:

- EN1 asks NN1 to find EN3 and determine what path through the network should be used.
- NN1 is not EN3's network node server so it needs to get help from the other network nodes.
- NN1 sends a request to all of its adjacent network nodes, looking for EN3. This is known as an APPN broadcast. NN2 is the only network node adjacent to NN1.
- NN2 passes the same request to all of its adjacent network nodes. NN3 is the only network node adjacent to NN2.
- Although EN3 has a link to both NN2 and NN3, NN3 is acting as its network node server. Even though NN2 knows where EN3 is, it will not reply on its behalf.
- NN3 asks EN3 what communications links it currently has. This information is important because NN1 must be able to determine all the possible routes through the network to get from EN1 to EN3. The ability to receive and respond to this request is another function the end nodes perform that LEN nodes can't.
- EN3 replies that it has links to both NN2 and NN3.
- NN3 tells NN2 "Yes, I have an LU named EN3" and passes along EN3's link information.

- NN2 passes the information back to NN1.
- NN1 now has to compute which of the two routes to get to EN3 is best. The methods for computing routes are described in the next section. NN1 passes the route it selected back to EN1.
- EN1 can now establish an APPC session to EN3 and start exchanging information.
- An application in EN1 can now establish an APPC session to EN3 and start exchanging information.

APPN networks get much larger than three network nodes. The broadcast that NN1 did to find EN3 would go to every network node in the network. Doing a broadcast every time a session needs to be started could use a considerable amount of network resources. To minimize the number of broadcasts, NN1 remembers where it found EN3. If another of NN1's end nodes asked to talk to EN3, NN1 would go straight to NN3 to verify that EN3 is still there and get any new link information from EN3. This *caching* of information is useful, because normally the end nodes attached to a single network node will be in some form of workgroup. People in a workgroup frequently run the same kinds of applications to the same destinations. Where this is true, network node broadcasts will seldom be needed.

Route calculations

The previous examples have used simple networks. For each session request, there were few possible routes through the network. APPN is capable of handling complex networks and providing intelligent, *class-of-service routing*. Class-of-service routing means that different types of data will each be routed using paths optimized for that specific type. All APPN nodes have several pre-defined classes-of-service, including "batch," "interactive," "batch-secure," "interactive-secure." Batch is used for sending large volumes of data—for example, file transfers. Interactive is normally used for programs that need quick responses. The secure class-of-service definitions give users the option of protecting special data when it is sent through the network.

Which class-of-service to use is determined by the *mode name* that is passed on the Allocate call. Given a class-of-service, APPN determines the importance of eight values that are defined for every link in the network. These include propagation delay, cost per byte, cost for connect time, effective capacity, and security. Each class-of-service assigns a different numerical importance to these values. For instance, when an application asks APPN for a batch session, APPN will try to find a path through the network that has high capacity and low cost. For an interactive session, APPN will try to minimize the propagation delay. This means avoiding links such as satellite links because of the time it takes to send a signal from the earth, up to a satellite and back. This delay could be enough to prevent acceptable performance. For secure sessions, APPN will only pick paths that are built from secure links. If no secure path is available, the session will not be started.

Also included in the decision is information about each intermediate network node along the route. Each network node maintains a value called its *route addition resistance.* By altering the default value, network administrators can select which nodes they prefer traffic to go through. Network nodes can also detect that they are under a heavy load and go into a *congested* state. This means that as long as they are congested, they will not be selected for new sessions through the network.

continued on next page

Advanced Peer-to-Peer Networking *(continued)*

A network node chooses the lowest class-of-service cost. It does this by first calculating the cost of each of the different routes to the destination it finds out about. This is not a dollar cost, but rather the result of a numerical calculation adding the weight of each node and link along the route being examined. If the lowest cost is offered by more than one route, one of them will be chosen randomly. This distributes the load across equal-cost routes.

Setting up a node

Because of the advanced features APPN provides, configuring a node can be as simple as the following:

End nodes:

- Provide your CP name (and optionally, any other LU names).
- Provide your network node's link address. This will allow the end node to bring up a link and automatically get a network node server.

Network nodes:

- Provide your CP name (and optionally, any other LU names).
- Provide link addresses to connect to any other network nodes.

With just this small amount of definition, nodes throughout the network can communicate with each other. All the necessary information is either dynamically exchanged or is determined from defaults. When you consider these savings at each node in your network, you realize that APPN can have a tremendous impact.

APPN products

Today, APPN capabilities are available on the following IBM products:

- OS/400
- OS/2 (with Extended Services or Networking Services/2)
- 3174 Establishment Controller
- System/36
- DPPX/370

Each of these products can be configured as an end node or a network node, except the 3174. (Because the 3174 does not run end user applications, it can only be configured as a network node.) More recently, APPN capabilities have been announced for these IBM platforms:

- VTAM
- RISC System/6000
- 6611 router

Find out more:
SNA INTEROP

When APPN was announced as part of SNA in 1991, four companies announced their intent to create end node products: Apple Computers, Novell, Siemens-Nixdorf, and Systems Strategies, Inc. The end node architecture is openly available in IBM publication number SC30-3422-2.

In 1992, IBM announced that it would also license the APPN network node technology. Four companies simultaneously declared their interest in exploiting network node functions in their products: 3Com Corporation, Network Equipment Technologies, Novell, and Systems Strategies, Inc.

Summary

Advanced Peer-to-Peer Networking is a major step in SNA's evolution to support distributed applications in customer networks. It allows large, complex networks to be built using many types of cooperating computers. APPN also offers dramatic reductions in the amount of network definition needed to configure each node in a network.

Acknowledgements

Thanks to the following associates for their careful review and comments: Mark Pozefsky, Anne Schick, and Wolfgang Singer.

Bibliography

IBM Manuals:

"AS/400 Communications: APPN Network User's Guide," IBM publication number SC21-8188. Describes the APPN support provided by the AS/400 system. Also describes APPN concepts and provides information for configuring an APPN network. APPN advanced considerations and configuration examples are included.

"Networking Services/2 Installation and Network Administrator's Guide," IBM publication number SC52-1110. Describes the APPN support provided by Networking Services/2 for OS/2 Extended Edition. Also describes APPN concepts and provides information for configuring an APPN network.

IBM Redbooks:

"APPN Architecture and Product Implementation," IBM publication number GG24-3669. Contains a tutorial on APPN, as well as an overview of the various product implementations.

"APPN/Subarea Networking Design and Interconnection," IBM publication number GG24-3364. A guide for planning interconnection of APPN and SNA subarea networks.

"AS/400 Distributed Systems Implementation Guide Volume 3," IBM publication number GG22-9458. Discusses the decision criteria that must be considered when choosing a topology for an AS/400 APPN network.

"S3/X and AS/400 APPN Nodes Using the SNA/LEN Subarea," IBM publication number GG24-3288. Describes the incorporation of a S/370 SNA subarea into a network comprising APPN network nodes. Intended for systems programmers and systems engineers in the intermediate systems and VTAM/NCP areas.

"Networking Services/2 Installation, Customization, and Operation," IBM publication number GG24-3662. Provides planning information for IBM SAA Networking Services/2. Contains an extended example on connecting Networking Services/2 and AS/400, with their respective configurations.

"3174 APPN Implementation Guide," IBM publication number GG24-3702. Provides guidance on implementing the 3174 APPN functions in various scenarios.

"AS/400 APPN with PS/2 APPN, 3174 APPN, 5394 and Subarea Networking," IBM publication number GG24-3717. Provides several scenarios of interaction of these nodes including sample definitions and traces.

STEVEN T. JOYCE manages part of the APPC Market Enablement team in the Architecture and Telecommunications organization. He has recently completed managing the planning and testing of the IBM SAA Networking Services/2 product. He joined IBM in Raleigh, North Carolina in 1983, where he was involved with the quality assurance and testing of IBM's office systems. He received a B.S. in Computer Science from North Carolina State University. sjoyce@vnet.ibm.com.

JOHN Q. WALKER II manages part of the team responsible for propagating knowledge about APPC, APPN, and CPI-C: open technologies for building powerful distributed systems. He recently managed one of the development teams responsible for the implementation of APPC and APPN for OS/2. He joined IBM in Rochester, Minnesota in 1978, where he was involved with the development and testing of the operating-system software for the IBM System/38. In Research Triangle Park, Dr. Walker was an architect for the IBM Token-Ring Network, serving as a co-editor of IEEE 802.5 local area network standards, 1983–1984. He received a B.S., B.A., and M.S. from Southern Illinois University and a Ph.D. in computer science from the University of North Carolina at Chapel Hill. johnq@vnet.ibm.com.

Ed.: An earlier version of this article has been published in the *IBM Personal Systems Technical Solutions,* January 1992, pages 67–72, IBM publication number G325-5014-00. See also "SNA Internetworking" in *ConneXions,* Volume 6, No. 3, March 1992.

The OSF Distributed Management Environment

by David Chappell, Chappell & Associates

Introduction

Building an effective environment for distributed computing requires creating an effective management environment. Managing a distributed environment is a non-trivial task, a fact that quickly becomes obvious to anyone actually trying to do it. All of the problems associated with managing the systems in the network exist, together with a substantial set of problems unique to the network itself. Sometimes categorized as *systems management* and *network management,* respectively, a complete management solution must address both these areas. The *Distributed Management Environment* (DME), the latest of the technologies produced by the *Open Software Foundation* (OSF), aims at providing solutions to the problems of managing distributed systems and the networks on which they rely.

Creating DME

A DME *Request For Technology* (RFT) was issued in July of 1990. This relatively short document defined the problems to be solved, most of which are described below. Anyone, OSF member or not, could respond. 25 organizations submitted technologies that were judged to be within the scope of DME, and OSF announced its selections from those 25 in September 1991. The choices included submissions from several organizations, including Hewlett-Packard, Bull, IBM, and Tivoli Systems. [1] Currently, OSF is integrating the chosen technologies into a coherent whole; the resulting source code can be licensed from OSF by anyone. DME Release 1.0 is scheduled to be made available by OSF in 1993.

One important caveat: as just stated, the integration of DME's various components is not yet complete. Some aspects of terminology or of the technology itself may change before integration is complete. The description of DME given here should be thought of as a snapshot of a quite stable, but not entirely finished technology.

Requirements for a management solution

Providing a complete answer to the problems of management requires several things, among them:

- A way to describe management information and a management protocol to get at that information;
- Management applications to make sense of the information acquired and to allow a network manager or system administrator to modify that information;
- Support services for management applications, along with appropriate application programming interfaces (APIs);
- A way for those applications to provide a consistent user interface, again including an API allowing access to it.

One way of viewing these requirements is pictured in Figure 1.

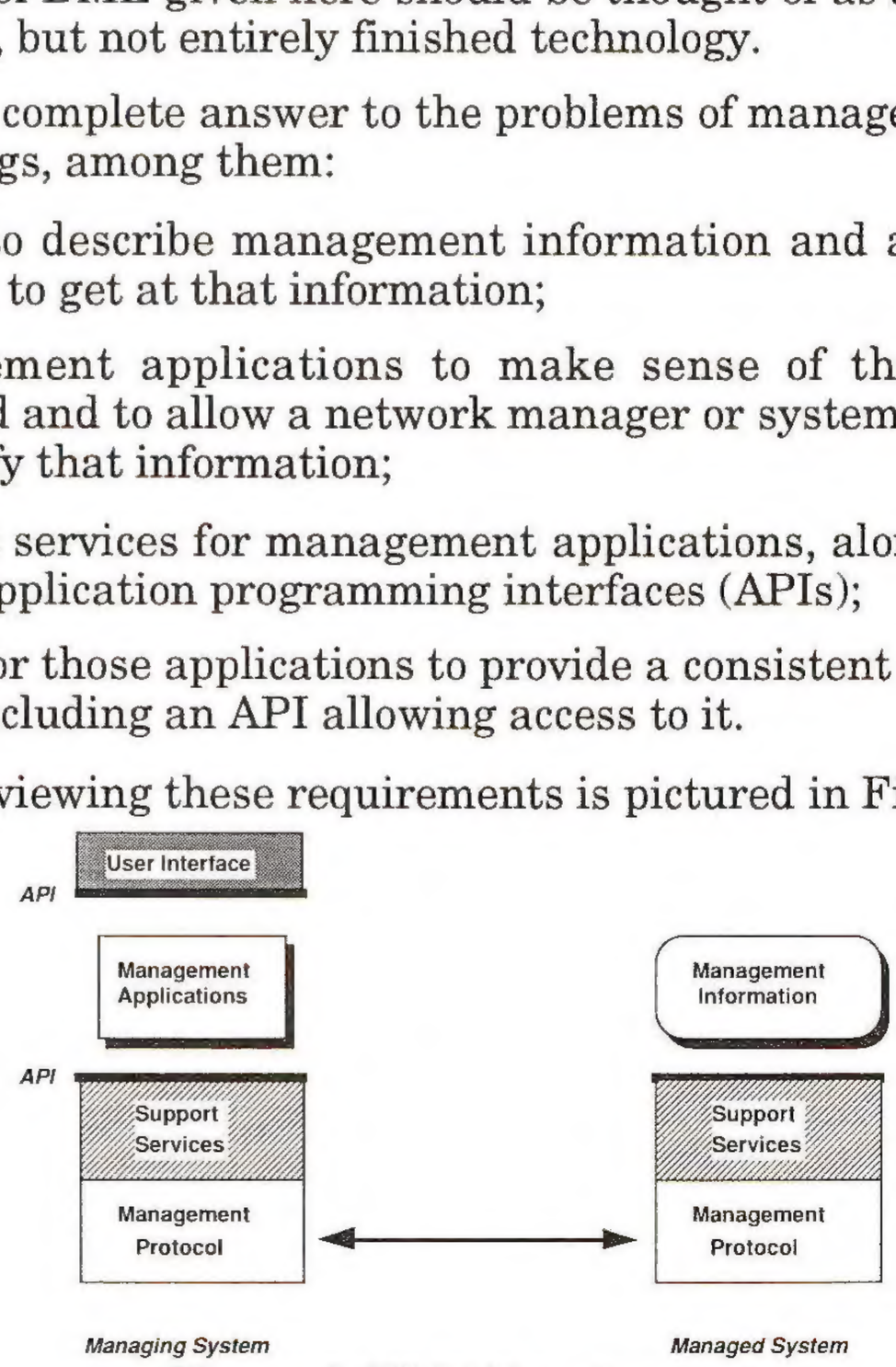


Figure 1: DME Requirements

Solutions to each of these problems exist today. Unfortunately, there has been no widely-accepted, vendor-neutral approach that solves all of them. DME's goal is to fill this void.

Meeting the requirements

Among the problems confronting DME's designers were:

- What management protocol(s) should be used to collect and modify managed information? The most widely used vendor-neutral protocol today is the *Simple Network Management Protocol* (SNMP). [2] SNMP provides an excellent solution for many problems, and its creation and quick rise to ubiquity have been a godsend for many. Still, its focus on network rather than systems management makes it an incomplete solution in the minds of all but its most rabid supporters. Another possible choice is OSI's *Common Management Information Protocol* (CMIP). [3] While no one can accuse CMIP of being too simple, it too suffers from its share of drawbacks. Chief among these is the current paucity of implementations; despite many years of standardization effort, relatively few products currently support the protocol. A second possible drawback is the protocol's size. Unlike SNMP, an implementation of CMIP does not fit comfortably in simple interconnection devices like bridges. And at least one more possibility exists, too, for accessing management information: a truly object-oriented solution, something that provides more than either SNMP or CMIP, especially for systems management.

- How should management information be described? Each of the management protocols just mentioned defines its own scheme for describing the objects that are to be managed, objects ranging from a variable that counts received IP packets to a table in a router to an entire computer system. These managed objects are specified and named according to the *Structure of Management Information* (SMI) of the chosen protocol. While the SNMP SMI and the CMIP SMI have some things in common, they also differ in some important ways. As might be expected, the SNMP SMI is simple, straightforward, and perhaps a little limited, while the CMIP SMI is none of those things.

- What kind of programming environment should be provided? Management applications need supporting services and a coherent, well-defined set of APIs. Many vendors have created their own platforms, including those in Sun's *SunNet Manager,* Digital's *DECmcc,* and Hewlett-Packard's *OpenView.* A vendor-neutral solution must provide a common answer to these problems.

- What should the user interface look like? A curse of multivendor networks is that each vendor's management applications provide a different user interface. While most sophisticated products today provide a relatively easy to use graphical approach, a human manager may still be faced with a plethora of differences in those interfaces. Ideally, all applications should follow the same set of conventions, providing a common look and feel for their users.

- And finally, perhaps most important of all: what kinds of applications should be provided? Without applications, all of the above components—management protocols, SMI, programming environments, and user interfaces—exist in vain. For OSF, chartered as a creator of "enabling technology," this is perhaps the most difficult question.

continued on next page

OSF/DME *(continued)*

One might argue that the correct answer is "None," that the job of an organization like OSF is to set the stage, then retire to the wings, allowing system vendors and independent software vendors (ISVs) to competitively play their parts in a now-open market. Alternatively, users might feel that their lives would be made much simpler by having only a single, standard way to perform important management operations. And what about applications not conventionally thought of as management applications, but that nevertheless can be useful as enabling technology? Some possible examples include a network print service or a backup and restore system. This kind of application might also be a useful part of DME.

The DME approach

Out of this surfeit of requirements, OSF has attempted to craft a coherent approach. The various solutions that comprise DME are grouped into two distinct parts. The first, called the *DME Framework,* provides just what its name suggests: a framework for management applications. This framework itself has two parts, one aimed at more traditional management applications, the other intended for more modern, object-oriented applications. The second part of DME consists primarily of solutions for some of the basic problems of distributed environments. This second part, called *Distributed Services,* includes a network print service, a network license service, a software installation service, and more. Figure 2 shows a simple taxonomy of the components of DME.

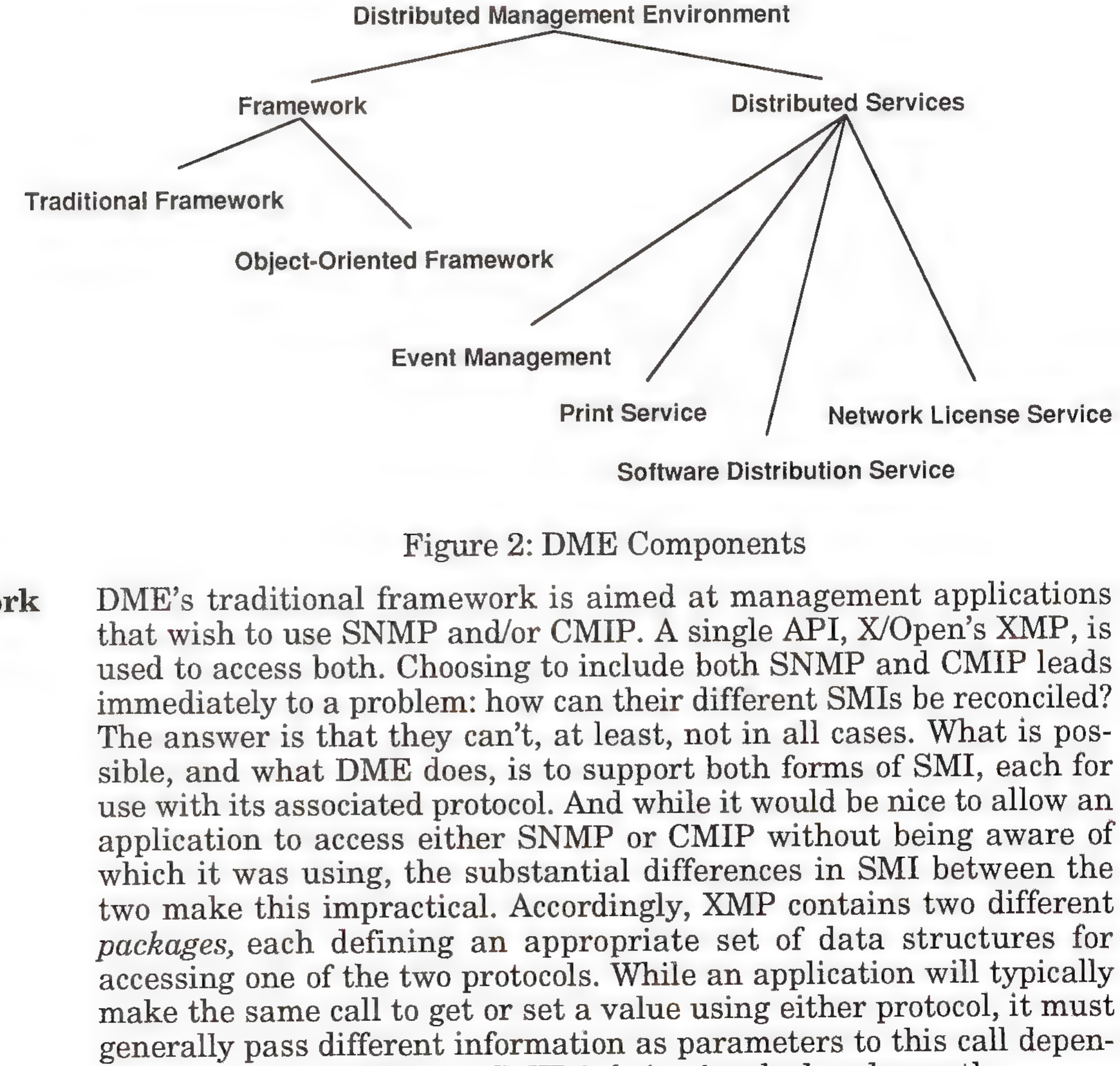


Figure 2: DME Components

The DME Framework

DME's traditional framework is aimed at management applications that wish to use SNMP and/or CMIP. A single API, X/Open's XMP, is used to access both. Choosing to include both SNMP and CMIP leads immediately to a problem: how can their different SMIs be reconciled? The answer is that they can't, at least, not in all cases. What is possible, and what DME does, is to support both forms of SMI, each for use with its associated protocol. And while it would be nice to allow an application to access either SNMP or CMIP without being aware of which it was using, the substantial differences in SMI between the two make this impractical. Accordingly, XMP contains two different *packages,* each defining an appropriate set of data structures for accessing one of the two protocols. While an application will typically make the same call to get or set a value using either protocol, it must generally pass different information as parameters to this call depending on whether SNMP or CMIP is being invoked underneath.

Figure 3 illustrates the components of DME's traditional management framework, giving some indication of how they are grouped into processes. (Note that DME will supply only the manager technology, i.e., the components shown in the middle box in the figure. Support for implementing SNMP and CMIP agents will not be included.)

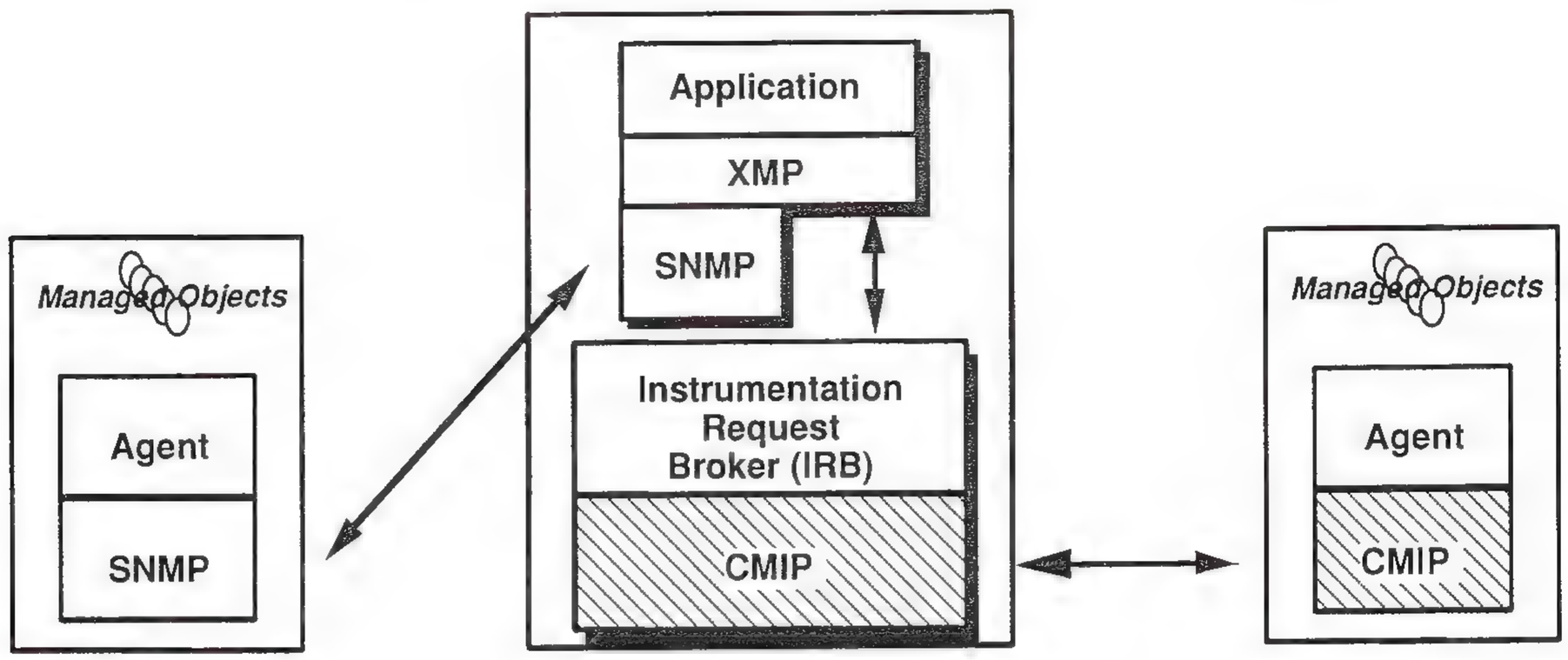


Figure 3: DME Traditional Management Framework

Objects DME's object-oriented framework takes a more aggressive approach to management technology. Aimed at systems management, this framework builds on work done by the *Object Management Group* (OMG), adding extensions where needed. The core of OMG's work so far has been the development of a *Common Object Request Broker Architecture* (CORBA). [4] CORBA defines how applications interface to an *Object Request Broker* (ORB), a system component that allows objects to invoke operations on one another regardless of their location. An example might be a payroll application that needs to access data controlled by some database management system. The DBMS might appear as an object to the application, one on which operations such as "query" can be invoked. Of course, the payroll application is also an object, and both objects communicate with each other via the ORB. Neither need be aware of where the other is physically located in the network.

This very general approach applies equally well for management. A management application (itself an object) can access managed objects throughout a distributed environment. This access is accomplished by invoking operations on those objects, operations that look to a programmer much like remote procedure calls. When an operation is received by its destination, an appropriate *method* is invoked to carry out the operation. In DME, a method can be implemented as a procedure, as a process, as a group of processes, or as almost anything else. One more point about the recipients of operations: they are themselves objects, and so can make requests of other objects. The client/server dichotomy of traditional management protocols need not exist in this object-oriented world.

DME's object-oriented framework supports all of this with a *Management Request Broker* (MRB), its version of an ORB. The DME MRB is a slightly extended CORBA-compliant ORB. The major components of this framework are shown in Figure 4 on the next page. As the figure indicates, MRBs on different systems communicate using the remote procedure call (RPC) mechanism of OSF's *Distributed Computing Environment* (DCE).

continued on next page

OSF/DME *(continued)*

While basing this communication on DCE implies the existence on managed systems of this rather complex piece of software, it also allows use of the many services implicitly provided by DCE RPC, including a name service for locating objects and extensive security features. And since the object-oriented framework is aimed at systems rather than network management, the burden imposed by DCE ought not to be too onerous.

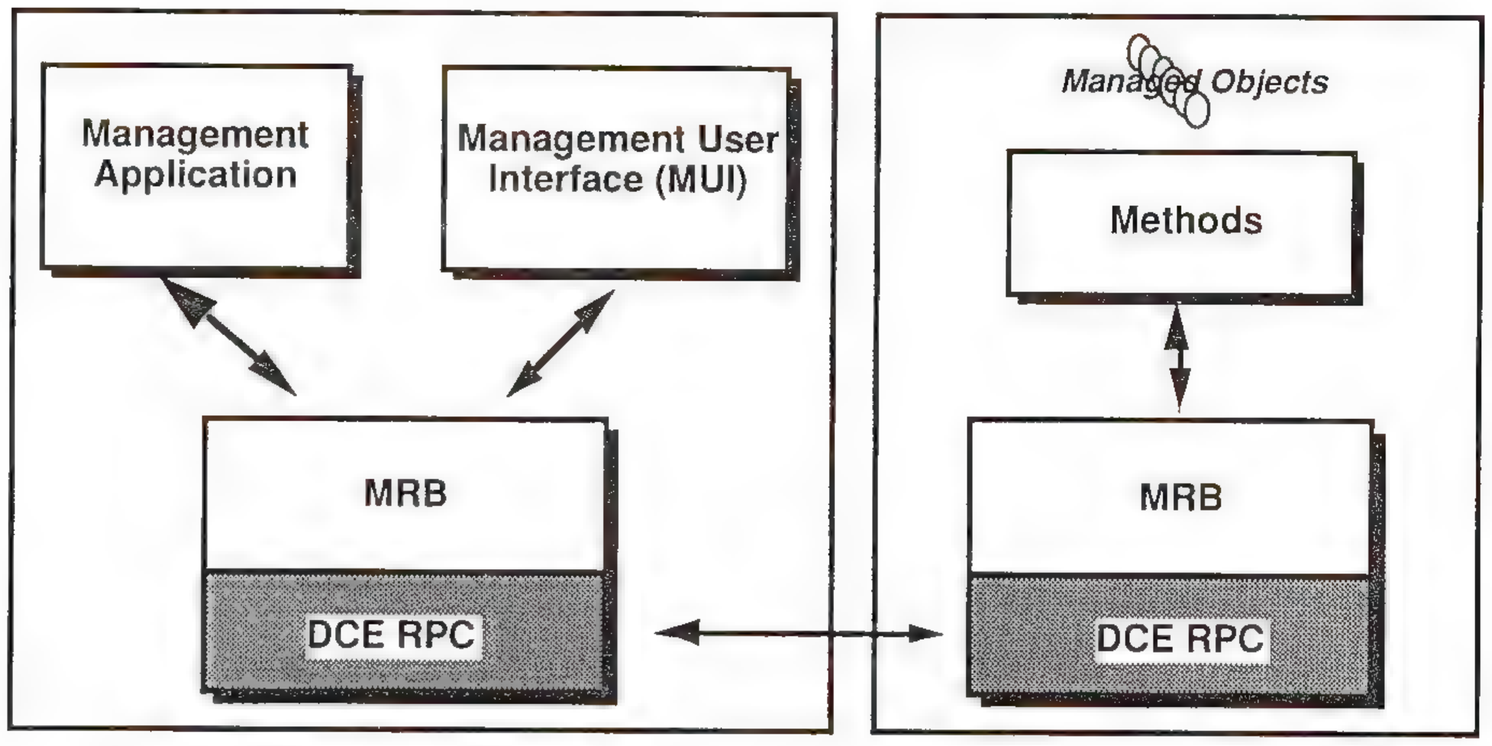


Figure 4: DME Management Request Broker

I4DL What about SMI? Both SNMP and CMIP go to some trouble defining a notation for describing managed objects. CORBA also defines a notation for specifying objects and the operations that can be invoked on them. Called the *Interface Definition Language* (IDL), it suffers from an unfortunate coincidence of names: OSF's DCE includes a wholly separate language with the very same name for defining the interface between RPC clients and servers. In DME, CORBA IDL is extended to allow specifying an object's inheritance, implementation, and installation properties, giving rise to a new notation called *I4DL*. A superset of CORBA IDL, I4DL adds extra functionality useful to systems management applications (and for that matter, to any other kind of applications—there is really nothing management-specific about I4DL or most of the rest of DME's object-oriented framework).

Two more aspects of the object-oriented framework are important to mention. One is the *Management User Interface* (MUI), a mechanism that allows many management applications to provide a common user interface. The MUI is just another object, one that makes available operations for affecting a terminal screen and invokes operations based on input from a human user. Based on The X Window System, it allows applications to display information to and receive input from a user in a common way. The other important aspect of this framework, one not shown in the figure, is the notion of an *adaptor object*. An adaptor object provides a way for things that already exist outside of DME's object-oriented framework to access and be accessed by DME objects. One example might be an adaptor object for SNMP. Its implementation might translate between SNMP's protocol operations and the operations expected by other DME objects, thus making SNMP directly available to those objects. Alternatively, an SNMP adaptor object might aggregate several SNMP operations on some managed object into one, providing a higher level of service to other objects in the DME environment. Whatever their function, adaptor objects are the key to incorporating non-DME things into DME's object-oriented framework.

Distributed services

Along with the DME Framework, OSF has elected to provide a few key distributed applications with DME. The major applications included are:

- Event services, providing a consistent way to log, filter, and forward events (such as errors) that occur in a distributed environment;
- A distributed print services application, based on the *Palladium* technology from MIT's Project Athena;
- A software installation and distribution application; and
- A network license service, providing software licensing services for applications in a distributed environment.

All of these application rely only on DCE. They have no dependence on either the traditional or object-oriented DME framework. In fact, DME's Distributed Services are so distinct from the DME Framework that OSF's current plans are to release the two separately. The source code for Distributed Services is targeted for release near the end of the second quarter of 1993, while the DME Framework release is expected near the end of 1993. And although by no means certain, it is certainly possible that each of these applications will eventually be licensed independently by OSF.

DME's prospects

DME is poised to become the standard platform for management applications in open environments. It has been endorsed by many leading system vendors, ISVs, and users, and OSF has already established a successful track record with its previous technologies. At least for the moment, DME is the only vendor-neutral solution that addresses the complete problem of managing distributed environments.

References

[1] Open Software Foundation, "Distributed Management Environment Rationale," September 1991.

[2] Case, J. D., Fedor, M., Schoffstall, M. L., Davin, C., "Simple Network Management Protocol (SNMP)," RFC 1157.

[3] International Organization for Standardization, "Common Management Information Service Definition; Common Management Information Protocol," ISO 9596.

[4] Object Management Group, "The Common Object Request Broker: Architecture and Specification."

[5] Millikin, M., "Distributed Computing Environments: Laying the foundation for the future of interoperability," *ConneXions,* Volume 4, No. 10, October 1990.

[6] Millikin, M., "Networks and Objects," *ConneXions,* Volume 5, No. 10, October 1991.

[7] *ConneXions,* Special Issue: Network Management and Network Security, Volume 4, No. 8, August 1990.

[8] *ConneXions,* Special Issue: Network Management, Volume 3, No. 3, March 1989.

DAVID CHAPPELL is principal of Chappell & Associates, a training and consulting firm focused on vendor-neutral distributed computing. Among his current projects, David is a consultant to OSF involved with DCE and DME. Before seeing the light, he spent several years working within the OSI effort.

The Simple Management Protocol and Framework

Managing the Evolution of SNMP

by
Jeff Case, SNMP Research
Keith McCloghrie, Hughes LAN Systems
Marshall Rose, Dover Beach Consulting
and Steve Waldbusser, Carnegie Mellon University

Introduction

The Internet-standard Network Management Framework has become deeply entrenched as the basis for interoperable solutions to the problem of managing networks. At the heart of this framework is the *Simple Network Management Protocol* (SNMP) which provides an effective means for monitoring and controlling heterogeneous devices in local and wide area networks. SNMP and related standards have become both *de jure* standards, as endorsed by the relevant standards body; plus *de facto* standards, as evidenced by the widespread support by vendors and popularity among managers of networks.

Background

The *Simple Management Protocol* (SMP) and its Framework represent the next generation, providing extensions to the initial version of SNMP and the entire Internet-standard Network Management Framework. Although the new Framework is substantively richer, it remains an evolutionary—not revolutionary—advance in the state-of-the-art. These extensions have been carefully engineered to preserve the highest degree of interoperability possible with products already deployed in existing networks.

The purpose of this article is to describe the motivations leading to the design of the SMP Framework, identify how it differs from its predecessor, and report on the current plans and status of efforts to standardize the new framework.

SMP and the corresponding SMP Framework improves and extends the initial version of SNMP and the other components of the Internet Standard Management Framework which include the *Structure of Management Information* (SMI) and the many documents which define the *Management Information Base* (MIB). SMP and the SMP Framework are direct descendants of SNMP and the Internet Standard management Framework, just as the SNMP Framework was a direct descendant of the *Simple Gateway Monitoring Protocol* (SGMP).

Ancient history

When SNMP was first defined in early 1988, it was one of multiple competing ideas. The other components of the framework, specifically the SMI and MIBs, were originally intended to be protocol independent and to be shared by multiple management protocols.

Additional work to extend and enhance the usefulness of SNMP has been an ongoing process. Substantial expansion of the Management Information Base through the definition of new MIB objects has been productive. A new administrative framework to strengthen the authentication, authorization, access control, and privacy aspects of the SNMP framework has been defined, implemented, and is nearing standardization. Many vendors and users have developed new and improved applications built upon the SNMP framework. Finally, researchers have published new accessories for use within the SNMP framework such as conformance macros which document an agent's capabilities.

However, to date, the SNMP community has not opted to evolve two of the components of the SNMP management framework: the SMI and the SNMP protocol itself.

These two components were first defined in 1988 and have not seen major changes or improvements since that time in spite of the fact that much new knowledge has been gained in the interim.

Politics

As the specifications of these components progressed through the IETF standardization process in 1989 and early 1990, there were voices calling for changes, changes which would "do something to stop the SNMP epidemic." These changes in the SMI, which would not have been backwardly compatible with the fielded SNMP implementations, were designed to injure the then burgeoning but still fragile SNMP market. The SNMP advocates were successful in resisting these undesirable changes only by resisting all changes, including known desirable ones. As a result, bad politics within the IETF standardization process prevented good engineering.

The four years of stable specifications has been a mixed blessing. One key observation is that the existing framework provides stable and effective network management of the Internet, which is used pervasively and continuously. It is exactly this stability that is one of the reasons for SNMP's unprecedented successes. However, this same lack of evolutionary changes has blocked needed changes and created pressures within the IETF to allow changes in the SMI and the SNMP. Consequently, in March of 1992, the leadership of the IETF invited proposals which address perceived deficiencies of the SNMP Framework.

SMP incorporates new evolutionary changes in both of these components, addressing most, if not all, of the perceived defects of the SMI and the SNMP protocol. These changes are evolutionary in that they are, to the maximum extent possible, backwardly compatible and do not contain any unnecessary or inappropriate changes.

Motivation and goals

The design of SMP was motivated by the same architectural principles that are key to the design of SNMP. Two of the most important of these are: minimizing the overall cost of network management by minimizing the cost of the components which are most numerous, and fostering operation robustness by locating control of management resources and the management function as closely as reasonably possible to the centers of responsible authority.

SMP was also designed to include backward compatibility whenever reasonably possible. In fact, many features of SNMP and the SMI are retained only in the interest of backward compatibility and to thereby ease the period of coexistence and transition which is expected to be lengthy. The changes were designed to minimize intrusiveness for implementors, for implementations, and for users of the management technology.

The work was timed so that implementors and vendors could produce one set of highly desirable changes, i.e., to implement other changes at the same time SNMP security is added to products. Users will similarly benefit from a single transition.

Design process

When it became clear that the IETF was going to solicit proposals for the next generation of network management, attempts were made to form the strongest team possible in order to prepare the strongest proposal possible. Several individuals gravitated toward one another. They possessed shared goals and values, strong experience with and understanding of the strengths and limitations of the existing management framework, and a willingness to invest many hours in the project. Each had made key contributions to the Internet-standard Network Management Framework.

continued on next page

The Simple Management Protocol *(continued)*

The team that emerged was:

Dr. Jeffrey D. Case, SNMP Research

Mr. Keith McCloghrie, Hughes LAN Systems

Dr. Marshall T. Rose, Dover Beach Consulting

Mr. Steven L. Waldbusser, Carnegie Mellon University

The community developing the standards for the initial version of the SNMP and its predecessor, the Simple Gateway Management Protocol (SGMP), has always followed a custom of tempering all proposals through implementation experience, and the development of SMP was no different. Because all four are designers and implementors, they felt it necessary to lock-step the SMP specification with implementation experience, just as these same individuals had led the efforts to convert the designs for SNMP security into working software.

As such, when the SMP specifications were published, there were four independent yet interoperable reference implementations—each written by a different author. Two of these implementations are freely available. Two are supported commercially. It is anticipated that these reference implementations will be used to accelerate the development of many SMP-based products, just as most of today's SNMP products are based on the reference implementations developed by these same authors during the design of the older framework.

The authors used multiple lists of issues to insure that the new design was thorough in its consideration of perceived problems and possible enhancements. These lists came from a birds-of-a-feather meeting at INTEROP 90, an open meeting of the SNMP Directorate of the IETF in the summer of 1991, frequently discussed issues on the SNMP mailing list, as well as the authors' personal experiences which added items others had overlooked. While the new design does not incorporate every suggestion nor ameliorate every criticism, each issue was given due consideration.

SMP features

There are many improvements found in the SMP Framework, far too many to list individually in this article. Eight of the most important improvements are described below:

Security and privacy

The SMP Framework builds on the recent work to enhance SNMP security. That work, as described in RFC 1351, 1352, and 1353, produced proposed standards for authentication, authorization, access control, privacy, and proxy relationships. The SMP Framework builds on this foundation. However, the SMP Framework incorporates several changes. First, the requirements for ordered delivery are relaxed. Second, the clock synchronization algorithm is simplified. Third, the requirements for DES are relaxed in recognition of export restrictions. Fourth, the requirements for access control at the instance-level were relaxed, in recognition of the heinous performance impact. The new SMP design now incorporates advisory locks across multiple managers. Finally, the configuration of the initial parties was changed to correct errors and to reflect other enhancements.

These new SMP features will allow managers of networks to incorporate network control operations with confidence. They will also remove the specious excuses that vendors have been using to explain their failure to implement the *set* operator.

Bulk retrieval

A small change in the protocol has resulted in major improvements in the ability to retrieve large quantities of data. This is one of the most exciting new additions. A new protocol data unit (PDU), the awesome *GetBulkRequest,* may be used to request the transfer of a potentially large amount of data, including, but not limited to, the efficient and rapid retrieval of large tables such as large routing tables or a bridge's forwarding database. One experiment used the SMP get-bulk operator to retrieve in excess of 9,300 table elements per second.

New data types and textual conventions

The new SMP Structure of Management Information (SMI) defines a few new data types for use with SMP. They include 64 bit counters which may be used when 32 bit counters overflow too rapidly to be useful, bit valued strings which may be used for enumerated bit strings, and the *NsapAddress* type which is useful with MIB definitions in OSI networks.

The textual conventions document defines several textual conventions via a new macro designed for that purpose. Two are noteworthy. The *TestAndIncr* textual convention can be used for defining local and/or global advisory locks on data structures to arbitrate between interactions from multiple management applications, potentially on multiple distributed management stations. The *RowStatus* textual convention is similar to a convention described in the RMON MIB for row creation, modification, and deletion. A new display clause in the macro will be helpful for management stations by providing hints as to how values can be displayed.

Richer error codes and exceptions

The SMP protocol incorporates a richer set of error codes and defines three new types of exceptions. These provide a richer collection of diagnostics which disambiguate the previously limited set of error returns. In the past, when a management application requested an agent to alter a particular variable, the agent sometimes refused and responded by saying "No!" With SMP, the agent responds by saying "No, because..." thereby providing the management application more information about the nature of the failure so that it can determine if the failure is permanent or temporary as well as attempt some corrective action.

Attempts were made to incorporate features into the design which reduce the number of times a manager station attempts to retrieve information but receives none because of an error condition. To that end, exceptions are now returned on a per-variable-binding basis in addition to errors on a per PDU basis. Under certain circumstances, partial results are returned with the exceptions flagged rather than rejection of an entire request.

In addition, attempts were made to reduce the number of conditions which result in a request being dropped by an agent silently, resulting in timeouts at the manager station. This is especially important as new security features will result in additional timeouts, such as result of party configuration errors. MIB variables have been added to instrument the remaining sources of silently dropped packets, in order to aid in diagnosing problems.

Improved sets

SMP improves the operation of control functions via the *set* operator in several ways. The authentication, authorization, and access control enhancements have already been described, as has advisory locking between management applications. In addition, SMP clarifies the procedures which may be followed in order to add and delete rows from tables. This has been a source of frequently asked questions and complaints in the past.

continued on next page

The Simple Management Protocol *(continued)*

Multiprotocol environments

SMP has several features which eliminate many of the TCP/IP biases of its predecessor. Transport mappings are specified for multiple protocol stacks, including TCP/IP, AppleTalk, IPX (Novell NetWare), and OSI, with provisions for others to be specified in the future as necessary. Support for new address types is also included.

The trap PDU, used for event notifications between entities acting in agent roles and entities acting in manager roles, has been redefined. The SNMP trap could only be used with TCP/IP networks because the only address type allowable in the PDU header was a TCP/IP address. The SMP corrects this problem by redefining the *trap* PDU which now has an identical form as the other PDUs, resulting in slightly reduced implementation size.

Manager-to-manager communications

The original SNMP architecture has supported manager-to-manager communications since mid-1988, even if few products implemented those features. SMP builds additional support for manager-to-manager communications in two ways. First, an initial MIB has been defined for this purpose. It is anticipated that additional MIBs will evolve as experience is gained. Second, there is new protocol support for persistent traps between two management stations for so-called "reliable" event notifications. Perhaps most importantly, these features draw attention to the fact that the capability for hierarchical management exists.

New SMI macros

The SMP SMI defines five new macros: object type definitions, object group definitions, module compliance definitions, agent capabilities definitions, and trap definitions.

Object type definitions are improvements upon the work first described in RFC 1212 (Concise MIB format). The ACCESS clause has been replaced with a MAX-ACCESS clause to make it clear that it specifies the maximum which makes "protocol sense." A new UNITS clause has been added. "Optional" has been removed from the the STATUS clause in order to eliminate the need for many spirited discussions in MIB standardization fights. It has been replaced by more powerful alternatives through object groups and module compliance definitions.

Object group definitions are used to identify collections of related objects which form a unit of conformance. The macro is used to concisely convey the syntax and semantics of such a group. This information had previously been communicated informally through ASN.1 comments in MIB modules. The object group macro presents this information in a machine parse-able format.

Module compliance definitions are used when describing requirements for agents with respect to object definitions. As such, they define the minimum requirements for conformance and are specified in terms of MIB modules and groups.

The *agent capabilities* macro is an evolution of RFC 1303's *module conformance* macro. The agent capabilities macro may be used to describe, in a concise and machine parse-able format, which MIB modules, objects, and values are actually implemented by a particular agent.

Trap definitions are used to define trap messages, including defining the information to be conveyed. They also assign an OBJECT IDENTIFIER to each defined trap.

Coexistence and transition

As stated earlier, one of the design goals was backward compatibility whenever reasonably possible. This greatly eases coexistence and transition between SNMP and SMP because it allows continued use of all existing MIBs. Of course, many designers will want to update their MIBs to take advantage of the new features of MIB definitions. However, in the meantime, the old MIB definitions are fully compatible with the SMP framework.

An orderly and graceful evolution was given much thought. The specification of the SMP framework includes one document which is devoted to coexistence and transition issues. There are two primary approaches, bilingual systems and the use of proxy.

Bilingual systems implement both SNMP and SMP. For example, a bilingual manager sends SNMP queries to SNMP agents and expects SNMP responses. Similarly, it sends SMP queries to SMP agents and expects SMP responses.

Alternatively, proxy may be used to convert from one message format to another. For example, a new management station might support only SMP message formats. These messages might be converted to SNMP messages via a proxy agent in order to support communications with older agents.

Both of these approaches have been implemented and found to be workable.

Standardization efforts

Eight documents totaling in excess of 200 pages which fully describe the SMP Framework were first published in early July. The publication was timed to occur approximately two weeks before a "birds-of-a-feather" (BOF) session at the Twenty-Fourth meeting of the Internet Engineering Task Force (IETF) in order to provide participants an opportunity for informed comment.

The BOF was held on Wednesday, July 15, 1992. Approximately 220 users, vendors, and other interested persons were present. The SMP authors made a 90 minute presentation and then responded to questions for another 90 minutes. The meeting concluded with a discussion of what should become of the documents. There was strong agreement on several issues.

First, there was strong agreement that the SMP documents should become the basis for a new version of the Internet Standard network management framework. While some participants expressed sentiments that the documents do not address all of the perceived problems in the current SNMP framework, it was agreed that the differences of opinion were such that they could be resolved through the normal working group process, rather than through competing proposals.

Second, there was strong agreement that there needs to be a single transition from SNMP version 1 to SNMP version 2 rather than a multistep transition including secure SNMP. It is generally felt that a single transition will benefit vendors and users alike.

Third, there was strong agreement that an aggressive schedule for standardizing the next generation of SNMP technology is appropriate. To that end, it was proposed that a working group be formed and initiate work immediately to begin the standardization process. It was proposed that the group hold its first meeting in September with the hope of completing its tasks in November at the next IETF plenary.

continued on next page

The Simple Management Protocol *(continued)*

Finally, there was strong agreement that the name of the result should be called "SNMP version 2," in lieu of SMP. Several individuals expressed the opinion that retaining the familiar SNMP name will result in less confusion in both the short and the long term.

In the time since the BOF, the IETF leadership has begun to act on the growing community sentiment and implement a plan for standardization, including formation of the working group. As of this writing, it appears the working group will be under a more realistic deadline to conclude its work on or before the spring of 1993.

It will be a challenge for the members of IETF to avoid protracted debate in the standardization efforts leading to the next generation of the Internet Standard network management Framework. It is difficult to estimate the results of 200 or more individuals each wanting to change "just" one percent of the design, many of which are mutually exclusive changes.

In the meantime, implementation experience with SMP continues to grow and expand. The reference implementations have been distributed widely and are now being used to produce SMP manager station and agent products. It is expected that several vendors will be announcing and demonstrating their SMP-based products at INTEROP 92 Fall.

For more information

There are several sources of additional information for those who wish to learn more about the topics covered in this article. The primary reference is the specification of SMP. 8 documents which fully describe the SMP Framework are available via the Internet (see below).

The SNMP mailing list is one source of up-to-date information about SNMP and related issues. Subscription requests may be sent to the electronic mail address `snmp-request@psi.com`.

A relatively new no cost publication, *The Simple Times: The Bi-Monthly Newsletter of SNMP Technology, Comment, and Events* is devoted to SNMP (including SMP) management. It is published electronically and in hardcopy. Interested parties should send a note to `st-subscriptions@dbc.mtview.ca.us` for subscription information.

Interop Company offers a new tutorial developed by Case and Rose which deals with advanced network management topics. It builds on the background provided in an introductory first course developed by Case.

References

[1] Case, J. D., Fedor, M., Schoffstall, M. L., Davin, C., "Simple Network Management Protocol (SNMP)," RFC 1157, May 1990.

[2] McCloghrie, K., Rose, M. T., "Management Information Base for network management of TCP/IP-based internets," RFC 1156, May 1990.

[3] Rose, M. T., McCloghrie, K., "Structure and identification of management information for TCP/IP-based internets," RFC 1155, May 1990.

[4] McCloghrie, K., Davin, J., Galvin, J., "Definitions of Managed Objects for Administration of SNMP Parties," RFC 1353, July 1992.

[5] Galvin, J., McCloghrie, K., Davin, J., "SNMP Security Protocols," RFC 1352, July 1992.

[6] Davin, J., Galvin, J., McCloghrie, K., "SNMP Administrative Model," RFC 1351, July 1992.

[7] *ConneXions,* Two *Special Issues* on Network Management and Network Security, Vol. 3, No. 3, March 1989 and Vol. 4, No. 8, August 1990.

[8] Marshall T. Rose, *The Simple Book—An Introduction to Management of TCP/IP-based internets,* Prentice-Hall, 1990, ISBN 0-13-812611-9.

[9] Jeffrey D. Case, James R. Davin, Mark S. Fedor, & Martin L. Schoffstall, "Network Management and the Design of SNMP," *ConneXions*, Volume 3, No. 3, March, 1989.

SMP documentation

The SMP specification consists of eight documents:

"Introduction to SMP"
"Structure of Management Information for SMP"
"Textual Conventions for SMP"
"Protocol Operations for SMP"
"Transport Mappings for SMP"
"Management Information Base for SMP"
"Manager to Manager MIB for SMP"
"SNMP/SMP Coexistence"

These are available as Internet Drafts, which are used for "work in progress" documents. To get a copy via e-mail, send a message to `mail-server@nisc.sri.com` and in the body put:

```
SEND draft-rose-smp-*-00.txt
```

To get a copy via FTP, connect to `ftp.nisc.sri.com` and do:

```
cd internet-drafts
mget draft-rose-smp-*-00.txt
```

JEFFREY D. CASE authored and co-authored several management standards, including SNMP. He is the principal author of a leading vendor-independent reference implementation of SNMP-based agents and manager stations which form the core of many vendors' SNMP implementations. E-mail: `case@cs.utk.edu`.

KEITH McCLOGHRIE is an Associate Director of Engineering at Hughes LAN Systems, Inc. where he is responsible for the development of network management products. He is a member of the IETF's Network Management Directorate and has been an active member of the SNMP working group since its inception, involved in the development of many MIB specifications. He is a member of the IFIP Working Group 6.6 on network management, involved in the organization of the International Symposiums on Integrated Network Management. He gained his B.Sc. in Mathematics from Manchester University in England. E-mail: `kzm@hls.com`.

MARSHALL T. ROSE is Principal at Dover Beach Consulting, Inc., a California-based computer-communications consultancy. He spends half of his time working with clients, and the other half involved in self-supported, openly-available projects. Rose lives with internetworking technologies, such as TCP/IP, OSI, network management, and directory services, as a theorist, implementor, and agent provocateur. Rose received the Ph.D. degree in Information and Computer Science from the University of California, Irvine, in 1984. His subscriptions to *The Atlantic* and *Rolling Stone Magazine* are in good standing. E-mail: `mrose@dbc.mtview.ca.us`.

STEVEN WALDBUSSER is the Manager of Network Development at Carnegie Mellon University, where he is responsible for providing network management systems to manage a heterogeneous campus internetwork of 6000 hosts and 150 networks. He is a member of the Internet Engineering Task Force's Network Management Directorate and the author of the IETF's RMON MIB (RFC 1271) and AppleTalk MIB (RFC 1243) specifications. E-mail: `sw01+@andrew.cmu.edu`.

Internet 2000:

A Protocol Framework to Achieve a Single Worldwide TCP/IP/OSI/CLNP Internet by the Year 2000

by Richard desJardins, The GOSIP Institute™

Introduction

It is urgent for the moderates of the TCP/IP and OSI/CLNP Internet communities to put the divisions of the past behind us and join together to achieve a single Worldwide Internet. Why let a second "OSI/CLNP Internet" get started, even if the "TCP/IP Internet" is dominant? The only interests served by this would be those of the hardliners of both communities and the vendors who make money on confusion and unnecessary duality. This article presents a proposal for a single protocol framework for the two communities. This article is offered for discussion within the communities involved, i.e., TCP/IP Internet represented by the Internet Society/IAB/IETF, and OSI/CLNP Internet represented by US GOSIP/IGOSS. This article is aimed at the moderate leaders in both organizations with an urgent plea for action within the very short timeframe necessary to head off two Internets.

Preamble

It is time to put the TCP/IP and OSI protocol suites together into a framework that allows "anything over anything" interoperation. The following proposal describes a simple (i.e., easy to understand), workable (i.e., easy to implement), and pragmatic (i.e., win–win: satisfies legitimate primary objectives of both groups) approach to achieving this goal by the year 2000. Because the approach is based on integrating IP and CLNP (*Connectionless Network Protocol*) [3] Internets to provide a single combined Internet by Year 2000, we have titled this proposed protocol framework "Internet 2000" as a working title. Comments on the content and title of the proposal are welcome.

Action requested

We would like moderates of both sides to comment on this proposal to help "all of us" (i.e., moderates who believe in internet technology and who respect both the TCP/IP and the OSI/CLNP Internet communities) to improve the proposal so that it achieves the goal of describing the overall architecture for forging a single community out of two. We plan to continually revise the framework in light of comments received and evolution of events, and make it freely available. If this approach is successful, we will have a working framework that both IAB/IETF and GOSIP/IGOSS moderates can relate to. The goal of this proposal, is to describe a single Worldwide TCP/IP/OSI/CLNP Internet that satisfies the objectives of both the TCP/IP Internet group and the OSI/CLNP Internet group.

Approach

The approach recommended by this proposal is to build Internet 2000 on the basis of the TCP/IP Internet and US GOSIP de facto four-layer architecture, which in this proposal is called the "Internet 2000 Framework": Application Services over Transport Services over Internetwork over Subnetworks. In this proposal, we use US GOSIP as a surrogate for IGOSS, which is the forthcoming *Industry Government Open Systems Specification* that will (hopefully) be compatible with GOSIP Version 3 and the *European Procurement Handbook for Open Systems* (EPHOS). This Internet 2000 Framework is a descriptive architecture that leads to a prescriptive set of progressive steps and interface definitions facilitating "anything over anything" interworking. Hopefully, this Internet 2000 Framework may approximate what is actually happening anyway based on the views of the moderates in both communities. Unfortunately, the moderate view often stands a real chance of being derailed by the hardliners of both sides who want to see "everyone" adopt their specific view.

The TCP/IP Internet hardliners believe their view is correct because it is based on *practical experience* and an *installed base.* The OSI/CLNP hardliners believe their view is correct because it is based on International Standards and on accommodating all consensus requirements. Fortunately or unfortunately, we believe they're both somewhat right, i.e., they're both right enough that their views should be respected and forged into a single working framework that satisfies the needs of both communities. That is what this proposal attempts to do.

Subnetworks Layer

The existing TCP/IP Internet does not specify subnetwork technologies. The TCP/IP Internet approach in the subnetworks area is to allow organizations to decide their own subnetwork technologies based on their own criteria.

US GOSIP, on the other hand, recommends a specific set of subnetwork technologies: GOSIP Version 1 specifies IEEE 802.3 Ethernet, IEEE 802.5 Token Ring, and X.25 packet switched network; GOSIP Version 2 adds HDLC point-to-point links and ISDN digital telephone network; and GOSIP Version 3 will add FDDI fiber optic LAN, Frame Relay fast packet switched network, and maybe SMDS connectionless cell switched network and PPP point-to-point protocol. US GOSIP requires agencies to focus on this specific subset of subnetwork technologies in order to promote interworking, competition and low prices.

Thus the two communities are already compatible at the Subnetworks Layer, in the sense that TCP/IP Internet current practice allows US GOSIP to decide its own subnetwork technologies based on its own criteria. Therefore this Internet 2000 Framework proposal for the Subnetworks Layer recommends that both communities continue their current practices.

Internetwork Layer

TCP/IP Internet currently requires IP as the internetwork protocol. IP is now being reworked to provide more addresses and to help solve the large flat routing table problem. A number of methods have been proposed. This Internet 2000 Framework proposal for the Internetwork Layer recommends that two of the methods being considered by the Internet *Routing and Addressing* (ROAD) Group are compatible with achieving a worldwide internet "dialtone": TCP over CLNP, (TUBA), and Class C supernets (CIDR). These two methods are not mutually incompatible but simply provide two evolutionary ways of dealing with IP address depletion while still providing traditional IP and CLNP dialtone. Neither of these two methods by itself deals directly with the routing table problem, but TUBA does allow the routing problem to be solved within the CLNP framework, as described below. A third method being considered by the ROAD group, IP over IP (IPAE), which does deal more directly with the routing table problem, would create a two-level IP address, which is a brand new scheme. This is a *Bad Idea,* because it creates a new two-level dialtone that would have to be developed and deployed, and that would keep the communities permanently divided. We don't need this (keep reading).

US GOSIP already requires CLNP. Functionally, CLNP is just IP with lots of addresses. CLNP solves the large flat routing table problem by having lots of addresses available, so that (in particular) end-systems may be assigned two (or more) addresses if they wish, at least one of which is pragmatically hierarchical worldwide.

continued on next page

Internet 2000 *(continued)*

Specifically, this Internet 2000 Framework proposal recommends that worldwide internet service providers (EBONE, NSFNET, regional internets, commercial internets) provide CLNP NSAP ddresses in a new Internet worldwide hierarchy to facilitate routing, while organizations may hide their own addressing structures behind gateway address mappings (or not, as they choose).

Transition

The above recommendations would produce a single Worldwide Internet that provides both CLNP dialtone and IP dialtone, with an imperfect but workable way (dual internetwork layer) of getting from one dialtone to the other. Some more details: Newer TCP/IP Internet hosts should be encouraged to implement TCP over CLNP in addition to IP. Older hosts should automatically be assigned a new Internet NSAP address for them to use whenever they decide to add CLNP capability. They could continue to use traditional IP dialtone into the foreseeable future, but the long-term view would be to phase it out after everyone has converted to CLNP dialtone as the "IP2000" protocol (7 years from now?).

Note that US GOSIP, IGOSS, and EPHOS also allow CONS (*Connection-oriented Network Service*). CONS is discussed in Annex A, because it is not the main issue and should not be allowed to be used to divert the attention of the moderate community from providing worldwide IP2000 dialtone to all groups.

Note that achieving worldwide IP2000 dialtone is the important part of this proposal, because it is the expensive part due to the large infrastructure investment that it represents. Once you have IP2000 dialtone, everyone new who joins Internet 2000 can support two (or more) Transport stacks for 50 cents worth of software as a way of transitioning to a single Transport stack by Year 2000 (see Transport Services Layer, in the following section).

Implementation

Operationally, this IP2000 Internetwork Layer proposal would be implemented within the existing Internet. (Note that CLNP and its associated routing protocols are based on and therefore improve upon IP experience and lessons learned. These protocols have already been implemented by the major vendors, and are currently being deployed in the major Worldwide Internet provider networks. We already know how to do global longest-prefix routing, which is the basis of IP and CLNP routing protocols. That's why it would be terribly unfortunate to develop and deploy a new IP-over-IP scheme.) Addresses would be recognized within US GOSIP and the other worldwide OSI bodies, but would be assigned by Internet. Internet would define *ping* and *traceroute* for CLNP, and would implement a dual *Domain Name System* to handle CLNP name-to-address resolution. The worldwide backbone and regional networks would define a routing hierarchy and addressing structure to solve the routing table problem by assigning administrative authority identifiers and routing domain addresses to regional networks, and routing domain and area addresses to organizational networks. The method of using hidden organizational addresses within domains and areas would be worked out by IETF in cooperation with ANSI X3S3 and ISO SC6.

Win–Win

The TCP/IP Internet community should be happy with this proposal (i.e., they should be able to say "we won") because it achieves a single internetwork dialtone worldwide which is completely based on IP functionality.

Future long-term solutions developed within the Internet community would be compatible with the entire Worldwide Internet, not just part of it. The US GOSIP/IGOSS community should be happy ("we won") because this proposal achieves the single dialtone using the GOSIP/IGOSS mandatory protocol CLNP. And the ISO/CCITT community should be happy ("we won") because this proposal is based on ISO/CCITT International Standards, including the worldwide OSI Network Service and NSAP Addressing Plan defined in International Standard ISO 8348, and uses the CLNP protocol defined in International Standard ISO 8473 [1].

Transport Services Layer

TCP/IP Internet currently requires that Transport Services be provided by TCP and UDP, i.e., these are the services provided by TCP and UDP protocol entities to their users in the layer above. There are no formal service specifications. The Sockets and TLI interfaces are de facto standard program interfaces to the Transport service.

US GOSIP requires that TP4 be used to provide the Connection-oriented Transport Service, and allows the use of CLTP (Connectionless Transport Protocol, equivalent to UDP) to provide the Connectionless Transport Service as an option [4]. These services are formally defined in ISO 8072 (= CCITT X.214). The X/Open XTI interface (which is essentially the same as TLI) is the de facto standard program interface.

Three parts

This Internet 2000 proposal for the Transport Services Layer is a three-part recommendation. First, TCP/IP Internet should continue to provide TCP and UDP services at Sockets and TLI interfaces, and US GOSIP should continue to provide the OSI connection-oriented and connectionless Transport services at XTI interfaces using TP4 and CLTP. (Note that the application program interface (API) to Transport Services is an application portability issue rather than an interoperability issue. Also note that the use of TP0 or TP2 over CONS is discussed in Annex A.) This first part of the recommendation simply means that both TCP/UDP and TP4/CLTP Transport protocol suites should be allowed in the near term. The type of Transport protocol entity bound to each NSAP address would be identified in the X.500 Directory. Transport addresses (= two-byte T-selector plus Network address) are already structurally equivalent between the two communities.

Second, both communities should recognize the legitimacy of the RFC 1006 [2] TCP/OSI Coexistence Stack (i.e., TP0 over TCP), and should carry the concept a step further by defining all the ways that Application Services from either suite may call upon Transport Services from the other suite (we call this "anything over anything" working). For all three Transport stacks, a Transport entity that wants to reach an NSAP address bound to a different type of Transport stack may still be able to interoperate if it knows the NSAP address of an appropriate Transport switch (called a "Transport bridge" within Internet and a "Transport interworking unit" within ISO). The IETF, in collaboration with ANSI, IEEE and ISO, should develop and specify the legitimate ways of calling upon and interoperating among these Transport stack combinations in the near term. IEEE POSIX is currently developing a *Detailed Network Interface* (DNI) that supports both Sockets and XTI (i.e., an application can run directly over one or the other Transport interface) as well as a *Simple Network Interface* (SNI) that hides the details of the Transport interface. This work should be accelerated and brought to completion, and its use should be recommended by both communities.

continued on next page

Internet 2000 *(continued)*

Third, for the long term (7 years?), both communities should work together to develop the next-generation Transport protocol. Note that TCP and TP4 are both based on the same generation of technology, i.e., the 1970s. TP4 is more efficient and faster than TCP with checksum turned off, but is slower in actual operation due to the choice of a heavy duty mandatory checksum. By the year 2000, rate-based and selective-retransmission Transport technology will be needed to run over high-speed, mostly-reliable networks such as Frame Relay [10], SMDS, and B-ISDN [7]. There is no reason in the world to perpetuate the "two-community" divisiveness into the next century. The two groups should join together, with either one taking the lead or both working together over the Internet, to define a single "TP2000" protocol, whether it be called TP2000, TP5, TPX, XTP, or whatever. In fact, why not use this problem as an opportunity, by using it as a pilot project to find out how the IETF, X3S3, and SC6 should work together in the future? Let's put the drafts out as RFCs, encourage comments and work over the TCP/IP/OSI/CLNP Internet (i.e., dual protocol hosts allowing access by both communities), and learn how to work together by actually working together? The two groups can "just do it" if they really want to. This pilot proposal for doing a combined IETF/ANSI/ ISO standard really is a perfect test case of "can't means won't." For all the IETF nay-sayers, we say, "Just try it! You guys just say how you'd like to work, and see if the ANSI/ISO process won't accommodate the best way of working. The moderates on both sides should just work it out. Just do it!"

This Transport Services Layer component of the Internet 2000 Framework proposal is a win-win situation because both communities would continue to support their defined Transport services, so both would be happy in that respect. Application services written to a standard API or XTI specification would run over both stacks for 50 cents worth of software. And both communities would converge to a common next-generation Transport protocol by Year 2000. Then we will have a common "TP/IP 2000," and a single Worldwide Internet community up through Transport.

Application Services Layer

The TCP/IP Internet currently plugs its Application services such as FTP and TELNET directly into Transport services. The virtue of this approach is simplicity.

OSI/GOSIP uses a three-upper-layer stack to provide its Application services. Session Layer is used to provide dialogue control (primarily, graceful close). Presentation Layer is used to identify alternative encodings. Application Layer is built up in standard ways called "application contexts" using building blocks called "application service elements (ASEs)" (e.g., *Association Control Service Element* (ACSE) to do call control, *Directory User Agent* (DUA) *Service Element* to look up information in the X.500 Directory). The virtue of this approach is interworking flexibility, but at a cost of complexity. The future OSI Upper Layer Architecture now under development may become fully recursive above the Transport Service, i.e., basically one layer with a specified three-layer internal organization, like the OSI Network Layer is now.

This Internet 2000 proposal for Application Services Layer recommends that both communities keep their current methods of providing Application services. This means specifically that OSI would continue to look into how to streamline its upper layers, such as the recursive upper layers architecture and the OSI Skinny Stack (see page 51).

(The OSI Skinny Stack was defined by ANSI to run The X Window System over OSI, but is much more broadly useful: Session full duplex data transfer, Presentation encoding identification, and Application Association establishment and release, all in 2K bytes of code instead of 30K bytes.) To interwork with each other, both communities should use Application gateways (i.e., dual Application service implementations with mappings between them), and both communities should provide their Application services over dual Transport stacks. OSI Presentation addresses of the form (NULL, NULL, T-selector, NSAP-address) may be used by both sides to address applications seen through gateways. The advantage of using this form is that it is already an X.500 Directory attribute type.

Both communities should continue to roll out new Application protocols such as SMP/SNMP, PEM/MIME/SMTP, X.400-1988, X.435-1992 (PEDI), ODA/SGML, Distributed Transaction Processing, Knowledge Discovery ("Knowbot"), *WorldWideWeb* (WWW), *Wide Area Information Server* (WAIS). Here's a good idea: Let's do an "open RPC" between IETF and ISO SC21.

The ISO/CCITT standardization process should increasingly take account of the principles and methods of the Internet standardization process: electronic "Groupware" distribution, commenting, and rapporteuring of draft standards; the principle of "implement first, standardize second"; and providing specifications and software implementations at low or no cost. The Internet should continue to take on the task of discussing, developing and deploying the infrastructure needed to support new standards. Ultimately, the marketplace will decide what works.

APIs

To promote application portability and interworking, both communities should continue to support the development and use of consortia-defined or IEEE POSIX open systems environments, APIs and protocols. These include a bewildering variety of specifications and software such as OSF/DCE and UI/ONC/ATLAS RPCs, IEEE POSIX SNI and ACSE API, X/Open CPI-C, OMG ORB API, common Messaging API as recently proposed by Microsoft to XAPIA, and the X.500 and FTAM APIs being developed within IEEE. Both communities should continue to support standards for running TCP/IP Internet applications over OSI/CLNP Internet stacks, such as The X Window System over the OSI Skinny Stack, SNMP over CLNP, and NFS over CLTP or over a connectionless ACSE Skinny Stack.

Conclusion

The TCP/IP Internet and OSI/CLNP Internet communities may be brought together to create a single Worldwide Internet by Year 2000 by adopting the recommendations in this Internet 2000 Framework proposal. We have the singular opportunity right now, in 1992, to "just do it," after which the opportunity may be lost. Once lost, we will have the Internet equivalent of having half the world using one electrical power standard, and the other half using a different and incompatible electrical power standard, and trying to plug into each other. Who wants this? Only the hardliners and the special interests. By supporting the "anything over anything" working and the "IP2000 dialtone" which are at the heart of this Internet 2000 Framework proposal, applications will run over APIs and multiple Application Services implementations and through Application gateways, Application Services will run over multiple Transport Service protocol stacks identified by Internetwork addresses, and CLNP will become the IP2000 Internetwork protocol operating over any Subnetworks technology.

continued on next page

Internet 2000 *(continued)*

A next-generation high-speed Transport protocol, TP2000, will be developed jointly by the two communities. This would be capable of running over both CLNS or CONS (see Annex A), and therefore would be capable of running over any future Internetwork protocol developed by the Internet community in collaboration with CCITT and ISO. Some more details: "Well known N selectors" should be reserved for TP2000, TCP and TP4 (e.g., Nsel = 65 always means TCP, Nsel = 66 always means TP4, and Nsel = 67 always means TP2000). Internet Society RFCs should be developed that define how to run multiprotocol stacks over the various subnetwork technologies, similar to the specification for Multiprotocol Interconnect over Frame Relay Networks [11]. Other RFCs should specify how to make CLNP operate pragmatically in the Worldwide Internet, and how to interwork different Application Services and Transport Services stacks across the Worldwide Internet community.

Comments and support requested

Please send your comments to:

The GOSIP Institute
3009 Mission Square Drive
Fairfax VA 22031
Fax: 703-691-0797
Internet: rdesjardins@attmail.com
X.400: c=us,admd=attmail,pn=richard desjardins

Annex A: CONS

The proposal in the main part of this document focuses on the *Connectionless Network Service* (CLNS), which is functionally identical to IP with lots of addresses. CLNS is required as the mandatory service of US GOSIP internets, and is allowed as an optional service of European OSI (i.e., UK GOSIP and EPHOS). An alternative flavor of Network service, *Connection-oriented Network Service* (CONS), is optional in US GOSIP and mandatory in Europe.

Transport options

The dominant view within the US is that CONS is suited only for use between two Transport entities which are directly connected by a single connection-oriented subnetwork, e.g., two end systems directly connected over an X.25, ISDN, Frame Relay, or ATM subnetwork. The Transport protocol of choice is either TP4 or a new-technology protocol, but for certain subnetwork technologies, a simpler Transport stack may make sense (e.g., TP0 over X.25). For connection-oriented subnetworks, and using an appropriate Transport protocol, Transport over CONS offers the advantage of simplicity and better performance. For example, it is easy to provide isochronous service over ISDN and ATM, and not too difficult over Frame Relay if occasional data dropout is allowed (e.g., for videoconferencing).

As an example of a CONS application, Transport throughput of 0.5 Mbps over ordinary digital telephone lines (i.e., ISDN Basic Rate Interface) can be achieved with V.42bis data compression. The cost of this transfer (about 5 cents per Mbyte) compares very favorably with every other form of data transfer, such as CD-ROM over Federal Express. The only thing it can't beat is a "free" network (but even in that case, *someone* is paying on the order of 5 cents per Mbyte).

Thus it appears that CONS may indeed have a place as an optional stack in the networks of the future. The type of Network service (CLNS or CONS) available at an NSAP would be an attribute type associated with the NSAP address in the X.500 Directory. There would be no interworking between CONS and CLNS because there would be no need for it, since everyone would support CLNS. Most applications in end systems with direct access to connection-oriented networks (e.g., ATM to the desktop) would run over both stacks, with both NSAP addresses published in the X.500 Directory. Separating Transport protocol stacks and Network service types by NSAP address is easy to do within the CLNP addressing scheme, which provides a one-byte N-selector for this purpose. Note that by Year 2000, both CONS and CLNS may be subsumed into a generic Network service.

Internet Peace

But let's face it: all the above arguments notwithstanding, the bottom line of the CONS-versus-CLNS discussion is that making CLNS mandatory even when running directly over a single subnetwork connection adds only a small performance penalty at both ends. Therefore, if this small concession is the price that it would take to make the TCP/IP Internet people happy, then by all means let's consider having CLNS as the one and only Network service, or collaboratively defining a new *Unified Network service* (UNS). World Internet peace is worth it.

References

[1] "Information processing systems—Telecommunications and Information Exchange between systems—Protocol for Providing the Connectionless-mode Network Service," ISO 8473, March 1987.

[2] Marshall T. Rose and Dwight E. Cass, "ISO Transport Services on top of the TCP," RFC 1006, May 1987.

[3] Hagens, Rob, "Components of OSI: CLNP or A Day in the life of Ivan CLNPacket," *ConneXions,* Volume 3, No. 10, October 1989.

[4] Hall, Nancy, "Components of OSI: The Transport Layer," *ConneXions,* Volume 3, No. 7, July 1989.

[5] *ConneXions,* Volume 3, No. 8, August 1989, "Special Issue: Internet Routing."

[6] *ConneXions,* Volume 5, No. 1, January 1991, "Special Issue: Inter-domain Routing."

[7] *ConneXions,* Volume 6, No. 4, April 1992, "Special Issue: Emerging Broadband Services."

[8] desJardins, Richard, "*Opinion:* OSI is (Still) a Good Idea," *ConneXions,* Volume 6, No. 6, June 1992. (See also page 43).

[9] Rose, Marshall, Comments on: "*Opinion:* OSI is (Still) a Good Idea," *ConneXions,* Volume 6, No. 8, August 1992.

[10] Kozel, E., "The Cisco/DEC/NTI/StrataCom Frame Relay Specification," *ConneXions,* Volume 5, No. 3, March 1991.

[11] Malis, Andrew, "Multiprotocol Encapsulation over Frame Relay," *ConneXions,* Volume 6, No. 8, August 1992.

RICHARD desJARDINS is education director of The GOSIP Institute. He is two-term immediate past chairman of ISO Subcommittee 21 on OSI and he was one of the original key contributors to the OSI Reference Model. He has served as a senior systems engineer with NASA, DARPA, and CTA Incorporated for over 25 years. He has an M.S. in computer science from the University of Maryland as well as degrees in mathematics and physics from Catholic University.

Moving Towards the Ultimate Telecommunications: Personal Communications Services (PCS)

by Joseph J. Potts, GTE Laboratories, Inc.

Introduction

High-quality, low-cost telephone service has been a way of life in the United States for decades. The development of *Personal Communications Services* (PCS) will put telecommunications technology on the brink of an expansion that may eclipse much of the technology that has come before. The easiest way to understand PCS is to compare it with a past cartoon "science fiction" item—The Dick Tracy two-way wrist communicator. In essence, this small, lightweight personal communication device is what PCS will bring to every individual who desires the convenience of being in contact with anyone, anywhere.

Service levels

Personal Communication Services can be broken down into four fundamental service levels:

- *Home Service:* New PCS digital transmission techniques should match the quality of today's wired telephone service. The wireless PCS connection to the network would provide a range of untethered freedom expanded to the size of a football stadium or a small neighborhood. A fixed-base PCS unit could be connected to all of the normal telephone sets, answering machines, or personal computers via the regular wire in the home.
- *Traveling Service—Outbound:* Portable units used for your normal home service will be usable when you travel, like today's portable cellular service, a "phone booth in a pocket," but at a lower cost than today's cellular phones.
- *Follow Me Service—Inbound:* This level of PCS capability would allow the receipt of telephone calls while, at the request of the individual, the telephone network tracks the movement of the user. A call could be received whether you are just a few miles from home or across the country.
- *Customized Individual Services:* The telecommunications needs of each individual are different. This level of PCS capability would allow the user to customize services to meet their specific needs.

Compared to most telephone service today, PCS shifts the control from the calling party to the called party. The user, when called, will have the capability to screen calls and have the phone ring only for desired calls. Others can leave messages, or the user could re-establish the call at his/her convenience.

What will make PCS possible?

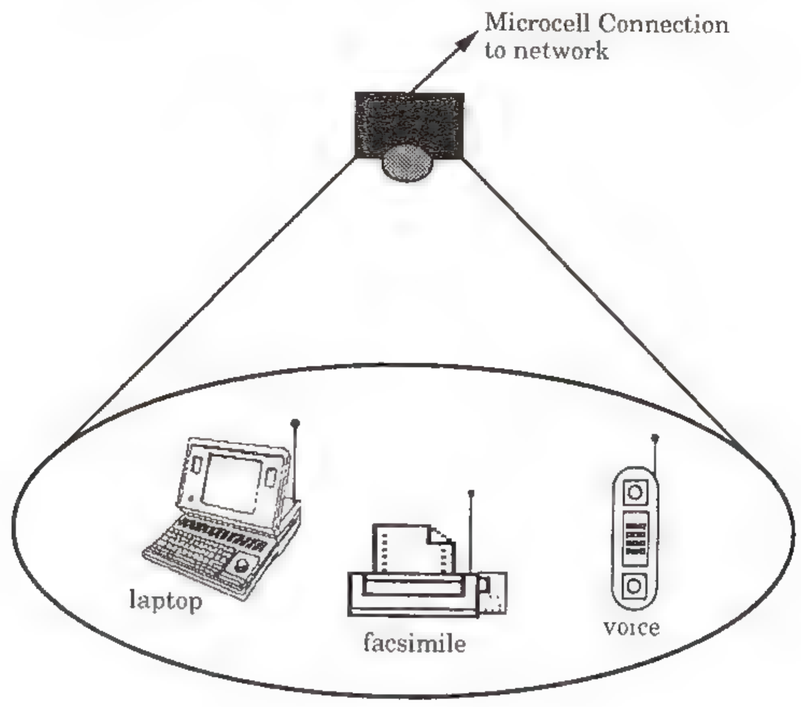
There are three factors that make PCS implementation credible:

1. New radio technologies are making the use of small, low-power radio transceivers, *microcells,* economically possible. PCS microcells provide orders-of-magnitude more capacity, and lower-power radios require smaller batteries, allowing for less weight in the PCS hand units.

2. The integration of computers into the telephone network provides a broad level of intelligence, which is needed to provide the dynamic routing capability for the PCS "Follow Me Service" and for "Customized Individual Services." The industry standards required for the use of network intelligence have reached "critical mass" and should be established soon.

Find out more:
Tutorial: T23
Session: G7

3. The consumer marketplace has shown a strong appetite for mobile services. Sales of cordless telephones are growing, and cellular telephone customers are being added at a phenomenal rate—in 1990 the cellular industry averaged 50 percent annual growth. This demonstrated market for mobile services provides stimulation for businesses to make the needed investments in research and development to implement PCS.



High-quality digital voice, data and fax are possible with wireless PCS microcell technology.

Limited to voice services?

The fact that a PCS network, sometimes referred to as a PCN (*Personal Communications Network*) will be a 100 percent digital and a mostly fiber optic network makes it highly suitable for data transmission as well as voice services.

Data connections on today's wireline telephone network typically are limited in capacity to 2.4–9.6 Kbps (kilobits per second); the PCS digital radio link could easily allocate 32Kbps per user, with some researchers feeling that 1544Kbps is possible. This factor would allow for accurate data transmission over the microcell radio link.

Challenges for PCS introduction

Several areas of concern need to be resolved to successfully implement Personal Communication Services. The major challenges are as follows:

Standards

Microcell standards are needed at the two interface points:

- Between the microcell radio base station and the fiber optic infrastructure.
- Between the microcell radio base station and the user's portable telephone radio.

This *common air interface* (CAI), will allow multiple manufacturers to design and build PCS portable telephones, and the resulting competitive market will keep unit prices lower, and unit functionality higher. The standard air interface will also allow users to shift easily between service providers.

Fraud and privacy issues

PCS network providers need a system that will minimize or eliminate fraudulent use of the network, (which is increasing significantly now in the cellular network). An on-line verification system is needed by the PCS network to perform the equivalent of a real-time credit check, like those done in retail stores.

Privacy and security are other issues to be resolved, since radio waves are relatively easy to tap into. Also, a great deal of information on the user becomes available for this level of personal communication.

continued on next page

Personal Communications Services *(continued)*

For instance, to provide the PCS user with the dynamic routing needed for call delivery and the database of user characteristics needed for custom service provision, a lot of personal information will be known by the network. The mobility tracking capability of PCS could have real "big brother" connotations unless safeguards are implemented into the PCS network from the beginning.

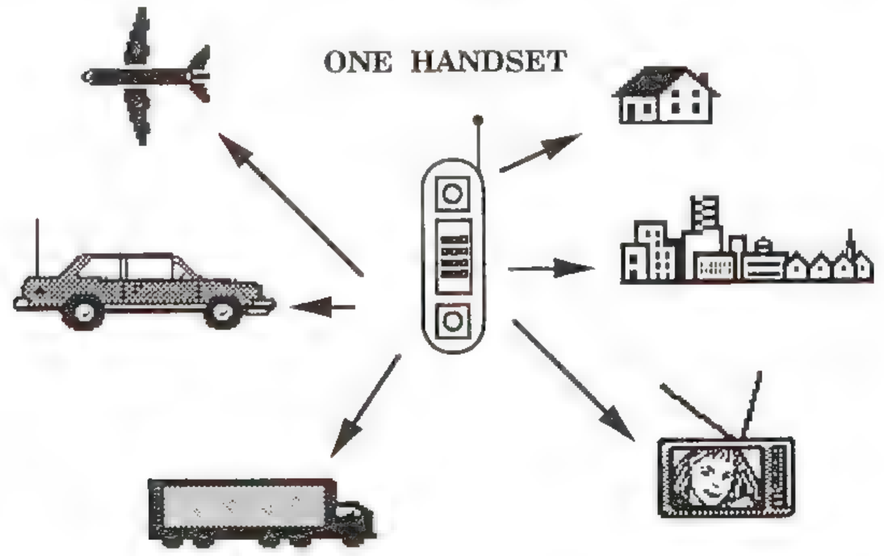
Human factors and ease of use

The intelligent network aspects of PCS will allow virtually unlimited potential for services while using the smaller, lighter hand units. The design of efficient and friendly user interfaces is needed to communicate the PCS user's desires to the PCS network.

Personal telephone number availability

In addition to the assignment of new telephone numbers to each individual who will be using PCS, the PCS user must also be able to reach any customer on the regular and cellular telephone networks. This requires that the numbering for the PCS user be compatible with the telephone numbering plan of the public switched telephone network (PSTN).

Availability of new numbers in the PSTN serving the U. S. is very limited, and they are geographical in nature. PCS telephone numbers would not necessarily have the same linkage to geography (like today's 800 numbers). A simple estimate of numbers required for PCS is one for every person! Insufficient Personal Telephone Numbers (PTNs) would stifle an emerging PCS implementation, and rapid introduction of massive amounts of PTNs could negatively impact current telephone users—either in convenience or cost.



Personal Communication Services Overview: The Year 2000.

The Future is now

There has been a tremendous amount of telecommunication industry interest in PCS over the past several months. A strong market for mobile services and the availability of new microcell and intelligent network capabilities seem to make the introduction of PCS a near-term reality. Cellular telephone operators currently have the flexibility to use their authorized radio spectrum for PCS, and some business announcements indicate PCS-like services will be available commercially within the next two years.

Telephone companies have made a rational case that a wireless link on the end of their existing network is a natural evolutionary step. Allocation of radio spectrum by the *Federal Communication Commission* (FCC) may be one possible stumbling block. Lack of much available radio spectrum makes this a difficult process for the FCC. It is believed that PCS will be available from multiple suppliers by the mid 1990s, and that it will be a major step in moving towards the ultimate telecommunications.

[Ed.: Reprinted by permission of GTE Laboratories, Incorporated from *ACCESS*, Autumn 1991 issue, pp. 12–13.

JOSEPH J. POTTS joined GTE in 1965, and is the Director of Program Development, GTE Labs. He is responsible for assisting the Vice President and Director of Research in structuring the research and development efforts for GTE Labs in strategic business support of the GTE Business Units. He has been at the Labs since 1986, after previously holding technology planning positions with GTE Telephone Operations. Mr. Potts received his BSEE from the University of Missouri and his MBA from Pepperdine University.

Book Reviews

Interconnections: Bridges and Routers, by Radia Perlman, Addison-Wesley, ISBN 0-201-56332-0, 1992.

Do us all a favor: Read this book!

At Last—a book that tells the "why" behind the bits and bytes of networking. In an insightful humorous way, Radia Perlman in her book *Interconnections: Bridges and Routers* pulls back the covers on the design of routers and bridges. As a real computer wizard, Radia can write about the deep mysteries of computer networks with sharp wit. Grab this book by any means necessary. Most of the these computer wizards are too busy generating the next generation of networks. Only rarely does one get the time to write down the "whys" behind the decisions that formed the networks we see.

For years I've admired Radia Perlman's work in standard committees. She and her peers at Digital Equipment Corporation, have contributed some of the best router and network design work to network standards groups. This book captures many of these design ideas behind DECnet Phase IV and DECnet Phase V. The book could be the best initial textbook for the novice or handy reference guide for experienced network person.

If this book were read by most people working with computer networks, computer networks would improve. Too often today, there is a lack of widespread knowledge of "what in the world" the standards people were doing when they designed bridges and routers and the protocols that glue them together. Without understanding *why,* people use routers or bridges in ways that break the networks. Very few people can describe "why" networks and bridges work, so most people have only read "How to" books on building networks out of routers and bridges. This "How to" knowledge tends to help one create networks which work most of the time but are poorly designed. By understanding why and how networks work, network operators may really build good networks.

Warning: Laughter may result

Radia's book can be hazardous to your reputation as a serious network person. I found that I laughed and laughed while reading this book. Since most of my reading took place late at night or on plane rides—this can be embarrassing. It is really tough to explain to the person sitting next to you that you are laughing hysterically about one of Radia's quotes such as "Algorhyme":

> *I think that I shall never see*
> *A graph more lovely than a tree.*
>
> *A tree whose crucial property*
> *is loop-free connectivity.*
>
> *A tree that must be sure to span.*
> *So packets can reach every LAN.*
>
> *First, the root must be selected.*
> *By ID, it is elected.*
>
> *Least cost paths from root are traced.*
> *In this tree, these paths are placed.*
>
> *A mesh is made by folks like me.*
> *Then bridges find a spanning tree.* —*Radia Perlman* (p 54)

or:

"The shortest distance between two points is usually under repair."

—*Anonymous* (p 265)

continued on next page

Book Reviews *(continued)*

For the network operator

Interconnections: Bridges and Routers walks the network operator through:

- *Link layer issues:* (such as Generic LANS, IEEE 802 LANS, 802.5, 802.3, and LLC)
- *Bridges and Bridge Algorithms:* (such as Transparent Bridges, the "Learning" Bridge, the Spanning Tree Algorithm, Why Transparent Bridges are really "Translucent Bridges," Source Bridges, SRT Bridges, comparison of Source Routing Bridges versus Transparent Bridging)
- *Network Layer Issues:* (such as connection oriented network layer issues with an X.25 example, connectionless network layer protocols in ISO and IETF, and network addressing in IP and ISO)
- *Routers, Routing Protocols, and Routing Algorithms:* (Network Layer Neighbor Greeting protocols—such as ES–IS, ICMP, and ARP; Network Routing algorithms such as Distance Vector vs. Link State; Routing protocols such as RIP, IS–IS and OSPF; Inter-Domain routing protocols such as EGP, BGP, and IDRP)
- Sabotage Proof Network Layer Protocols
- "To Route or to Bridge: is that the question?"

Value

By sharing how the standards committees created various protocols, Radia leads the reader into understanding what these standards really do. While *Interconnections: Bridges and Routers,* nicely lays out the bits and bytes of protocols for quick reference, the real value is the discussion of the shortcomings or benefits of each field. For example, the following two discussions on bit ordering in 802.3, 802.4 and 802.5 quickly highlight the "bit ordering" problem which plaques bridges:

"With 802.3 and 802.4 the least significant bit is transmitted first and with 802.5 (and FDDI), the most significant bit is transmitted first." This would not be an issue (adapters on receiver and transmitter for a particular LAN would presumably be symmetric, and the order of transmission would be irrelevant) except for the fact that the group bit in addresses was defined not as "the most significant bit" or "the least significant bit" but rather as "the first bit on the wire."

"The failure of the 802 committee to agree upon a bit ordering for all the LANS has caused immense amounts of confusion and interoperability problems." This type of concise description will quickly help the network operator understand and debug problems with media. Other descriptions will help the user decide between routers and bridges, different routing protocols within your network (IS–IS versus OSPF), and different bridge algorithms.

The descriptions of IS–IS and OSPF are so well written, that the network operator may take the descriptions and a network analyzer to start debugging his problems. As a network operator, you may enter into a new realm of debugging when people find you understand the IS–IS protocol. Don't tell 'em you read Radia's book and you'll leave an impression of how brilliant you are.

For the student

If you are a student, read this and ignore your professors. Beyond the theoretical is the harsh reality of "what works," Radia has both the ability to lecture you on the theoretical and the practical and to never stop asking "Why?" Her irreverent description of ISO network layers teaches you what the OSI model is really trying to do.

As a student, I found homework assignments to be challenging but dull. Radia's homework assignments radiate wit and fun. Many of them point up problems she's found in real life products. One especially wonderful homework assignment teaches you to translate English into "ISOese." I hope this book will form the basis of a new generation of textbooks for Computer Engineers and Programmers.

Sigh—not enough

My only problem with this book is what it covers in the area I work most in—inter-domain routing protocol. The text in this chapter of the book went to press 1 year ago, and a great deal of changes have occurred in the ISO, ANSI, and IETF work in inter-domain protocols. Radia only presents some of the current "whys" behind the current designs. I suppose the best comment on this is Radia's on closing:

"Well, I must somewhat regretfully end this book, both because many people would prefer to obtain it before the millennium and because my editor tells me its getting a bit long."

Inter-domain protocols

On the "whys of Inter-Domain Routing Protocols" Radia correctly observes that:

"The reason inter-domain routing protocols are different from intra-domain routing protocols is because it is assumed that routing domains don't really want to be combined into one big happy melting pot of a network. Routing Domains are independently funded. When a domain routes transit traffic, resources are being consumed." (p 313)

Thus a domain wants control over the traffic going across its links. However, other reasons that inter-domain routing is different are:

- *Clear Boundaries in protocols help delineate both the routing protocols, and debugging and operational issues:* The clear boundaries NSFNET has used in the connections on to the NSFNET backbone has been a help to isolating routing problems, and to delineate responsibility in debugging problems.
- *Policy is needed not only to satisfy political boundaries but to provide filter to dynamic routing information:* Radia notes that by law, packets cannot go between eastern Canada and western Canada by means of traveling through the US. The political and policy reasons are but one side of the nature of policy. Perhaps the rest is trying to make "good" and hopefully symmetric pathways out of the plentiful interconnections in the Internet. Pathways go up and down in the large Internet on a regular basis, but Inter-domain routing tries to allow the user to preselect what will primary and backup routes through the Internet for a network.
- *Try to reduce the amount of routing information that each network has to carry:* The IP routing tables currently grow at one entry per network. With very high growth rates, it is hard to keep enough memory on each router. Using hierarchy to summarize many routes into single table entry will help reduce the information. Beyond just grouping reachable networks in a hierarchical fashion, networks with common policies may be grouped.

Confederations

Section 10.2.4.2 on "Confederations" for IDRP describes how confederations work. While, the "why" behind confederations does not include the IP problems, the text does describe the benefits and costs of using Confederations to aggregate policy and routes. This section's description of the "hows" of using Confederations are excellent, and not available in the protocol standard.

—Sue Hares, Merit

continued on next page

Book Reviews *(continued)*

The Internet Message: Closing the Book with Electronic Mail, by Marshall T. Rose, Prentice Hall, Inc., ISBN 0-13-092941-7, 1992.

Rose is back. Despite claims to the contrary in his last book he has produced a *fourth* text in his trilogy on networking. *The Internet Message* (*Message*) is a comprehensive survey of electronic messaging technology used in the TCP/IP Internet today. Rose brings, perhaps, his strongest hands on perspective yet to the topic of electronic mail in the Internet.

Message exhibits the same consistent excellence of Rose's previous works. *Message*'s style lends itself to easy reading despite the technical nature of its content. *Message*'s message is clear—very rich electronic mail technology is here—and it isn't coming from X.400. Virtually all the advances and practical use of advanced messaging features from multi-media to active mailboxes are being worked within the Internet protocol framework.

Message digs down on the details of working e-mail systems in the Internet. In a previous review of Rose's work I disavowed the rumors that Rose was a secret agent sent to destroy OSI—well, times change, but I still don't think there is any secret to it. This is not a book that OSI supporters will enjoy—there are no flattering words for OSI found here.

Soapboxes

No review of a Rose book would be complete without reference to his trademark soapboxes. Throughout the text Rose gives insight to his personal perspective on controversial issues. Having been a consistent and early adopter of electronic mail technology Rose is able to provide many insights into the pitfalls of existing approaches and the controversies surrounding the issues and potential solutions. To those already familiar with electronic mail in the Internet the soapboxes provide important insight into the bigger issues surrounding the basic use of the technology.

Not the same Rose

Rose's style in *Message* takes a noticeable departure from his previous work. The style of Message is less technical and does not present hands-on programming examples. Instead the text focuses on the various components of technology and their relationships. In place of the traditional programming examples, *Message* presents detailed, and somewhat bizarre, syntax breakdowns of protocols using a BNF notation and then followed by specific examples of protocol usage.

Also, *Message* brings with it Rose's no holding back assault on OSI and X.400 in particular. Unlike previous works that have maintained an even handed perspective on the various technologies, *Message* takes a very pro-Internet and con-OSI perspective. Even the OSI directory, X.500, previously documented in Rose's *The Little Black Book* is viewed with significant skepticism.

Organization

Message begins with a general survey of electronic mail and the components of e-mail systems that will be discussed in the remainder of the text. Included with this introduction is the characteristically good preface by Rose. The next several sections involve review and discussion of issues surrounding naming, message formats and mail transports. Next *Message* turns to discussion of less widely used technologies including remote user agent facilities, multi-media mail and secure mail. The multi-media section focuses on the MIME standards recently developed in the Internet. The secure section focuses on long awaited the *Privacy-Enhanced Mail* (PEM) technology.

Finally, *Message* discusses more speculative issues, the thorny issue of mail gateways and then a high-level discussion of issues and developments in our electronic-mail future.

Included in the appendices are specific references to finding the available Internet standards (RFCs) and MIME software. Then the final words in *Message* are a reprinted paper from a conference where Rose provides his perspective on the future of OSI—a dismal future.

Value

The greatest value in *Message* is the bringing together of information on various electronic mail activities and technologies. Finding a single source of reference to topics such a basic SMTP and RFC 822 mail through the PEM and MIME work will be an invaluable asset to those working in these areas. Additionally, *Message,* is loaded with good examples of the matters discussed in the text and strong discussions surrounding the issues and the author's perspective.

Conclusions

The Internet Message is a "must read" for those involved with the electronic mail industry or research communities. It fills a noticeable gap in existing literature by covering the Internet perspective on electronic mail. *Message* is not a book that addresses lots of implementation issues—instead it plows through the issues relating to what variations on real implementation are good for. In fact, implementors looking for a heady technical discussion may find that *Message* falls short of Rose's previous works.

The bottom line—Rose's perspectives are not without controversy but he is one of the best authors in the industry—*The Internet Message* is worthy of prime shelf space for anyone involved in electronic mail (except those hanging onto the X.400 dream).

—Chris Moore, General Magic

Exploring the Internet: A Technical Travelogue, by Carl Malamud, Prentice-Hall, ISBN 0-13-296898-3, 1992. *[This book will be included with the INTEROP 92 Fall conference proceedings —Ed.]*

Great

It is indeed a pleasure to review *Exploring the Internet.* Of course, since you've already read two installments of this book ("Internet Explorer in Japan," May 1992, and "The Guy with the Bike," August 1992), you don't need to read this review. You already know that the book is going to be great.

What then remains to be said? Malamud's book reminds me of the style used by Douglas Adams' in his *Last Chance to See* (Harmony Books, ISBN 0-517-58215-5) in which Adams and zoologist Mark Carwardine get to circle the globe visiting endangered species. Along the way Adams both amuses and amazes with his description of current and past events and the people who take part.

Heroes and villains

So it is with Malamud's *Exploring the Internet.* As Malamud circles the globe, he finds out about current and past events in networking. Along the way, he gets to meet the heroes and the villains who are a part of those events. Forgive me for being crassly commercial here, but the best advice I can give you is to go to INTEROP Fall 92 and get the book. If you can't get to INTEROP, then call up Prentice Hall (they have an 800 number in the US) and order it. Read the book, you will be both informed and amused. Nothing more need be said.

—Marshall Rose

continued on next page

Book Reviews *(continued)*

TCP/IP Network Administration by Craig Hunt, O'Reilly and Associates, Inc., ISBN 0-937175-82-X, 1992.

As the Internet suite of protocols becomes an ever-more pervasive component of computing environments, so has the need for information on how to administer host systems which use these protocols. This book is a fine addition to O'Reilly's *Nutshell* handbooks as it provides a comprehensive hands-on guide in this area.

UNIX orientation

Actually, the title is somewhat misleading in that this book really focuses not on TCP/IP network administration but on the administration of hosts running TCP/IP. That is, if you're interested in administering your network (the routers, bridges, repeaters, and cables), then this isn't a book for you. Further, the kinds of systems being administered should run the Berkeley variant of UNIX, and not some other operating system, such as DOS or VMS. (Although this book tries to be "UNIX generic," its coverage of the AT&T variant of UNIX, System V, is limited to SVR3, not the current release SVR4.) With that bit of truth in advertising behind us, let's look at what this book does cover.

The book begins with three chapters containing the rote information on protocol models, architectures, layering, and so on. Towards the end of the second chapter, we are introduced to addressing and routing and how a user can interrogate UNIX for information as to how packets are forwarded from the host, while the third chapter explains naming, both in terms of the old centralized `HOSTS.TXT` file and the current *Domain Name System*.

Configuration

The next eight chapters all deal with configuration in one way or another: getting network numbers or domain names, configuring the kernel for TCP/IP operations, configuring interfaces to pass packets (including discussions of SLIP and PPP), configuring routing (including running a routing daemon), configuring the domain name system, configuring NFS, and, finally, treading where dragons fear...configuring *sendmail*. The upshot of all this is that this book tells you how to configure all the networking software on your host, from the link up.

Troubleshooting

A more interesting contribution, however, is the next chapter that deals with diagnosing networking problems. It is here where this book comes closest to the function described in its name, by showing how to analyze network problems from the host's perspective: connectivity, routing, and name service. Following this chapter is a brief discussion on security along with an introduction to various programs which perform auditing and access control.

Finally, this book wraps up with a discussion on how to find things in the Internet, a quick list of Internet service providers, and then some very lengthy appendices on how to configure the programs used for routing, naming, and electronic mail on Berkeley UNIX (*gated, named,* and the unfortunate *sendmail*).

Good reference

TCP/IP Network Administration makes a good reference work. It really doesn't contain any information that you couldn't find elsewhere, but it provides a nice overview of dozens of topics. As such, it is likely to be the book you reach for first; and if the coverage isn't adequate, then you can go to a specialized work, such as the *Sendmail Installation and Operation Guide*. Indeed, this book contains the pointers you might want to follow. Used in this fashion, this book is a fine addition to the host administrator's toolkit.

—Marshall Rose

Internetworking with TCP/IP, Volume III: Client–Server Programming And Applications (BSD Socket Version), by Douglas E. Comer and David L. Stevens, Prentice Hall, ISBN 013-474222-2, 1992.

This book is the third volume in Doug Comer's series TCP/IP. Volume I explained the TCP/IP protocol suite. Volume II primarily focussed on how to implement the protocols in an operating system. Now Volume III explains how to use the TCP/IP protocols to build your own distributed applications, using the BSD *socket* interface. (A version of the book using the less popular AT&T TLI interface is also planned.)

Best of three

Volume III is arguably the best of the three. The strength of Comer's books has been the perspective he brings as an experienced implementor and instructor. As an implementor, he knows that sometimes one has to worry about the theory along with where all the bits and bytes are, and as an instructor, he's learned when a reader is likely to want to learn theory and when the reader needs to learn about bits and bytes. So this book is for implementors, by an outstanding implementor-instructor (along with his erstwhile co-author, David Stevens).

Organization

The book is organized in a series of steps, with each chapter adding information to the previous one. The first five chapters introduce the client–server model, the concept of concurrency, and the BSD socket interface. Chapters 6 and 7 work through all the nitty little details of writing a client program (from how to locate a server, to designing the user interface) with example implementation of *daytime, time* and *echo* clients.

Chapter 8 starts eleven outstanding chapters on how to implement a server. It itemizes the detailed decisions an implementor needs to make (e.g., choosing between a stateful and stateless implementation, what addresses to listen at, and how to minimize response time). Then Chapters 9–18 elaborate on each topic. For example, Chapter 9 is on writing connection-less UDP servers (and shows how to write a *time* server). Chapter 11 describes how to implement a connection-oriented TCP server using multiple processes, while Chapter 12 explains how to use a single process (in both cases, using the *echo* server as the example server). Chapter 14 explains how to implement servers like *inetd* that support multiple transport protocols (both UDP and TCP) and multiple servers in a single program.

Having fully sketched out how to design a server, the authors then go on to explain the more complicated building blocks. Chapter 19 describes external data formats (in particular XDR) and chapters 20–22 describe RPC in detail, including how to use stub compilers.

The next four chapters are studies of particular implementations. Chapters 23 and 24 describe how to implement NFS (using XDR and Sun RPC). Chapters 25 and 26 walk through all the problems of implementing a Telnet client with copious examples of the required code.

Helpful hints

The text concludes with a helpful summary of hints implementing servers on a BSD system. Programmers will find these hints help them avoid security bugs and subtle failure cases in their implementations. In all, the book is a nice addition to any implementor's bookshelf.

Many readers are likely to be familiar with another book that covers much of the same material, Rich Stevens' *UNIX Network Programming*, and may wonder what the key differences are. Both are good books. The important differences are that Stevens' book has a broader scope (presenting both sockets and TLI), while Comer's book has deeper coverage of common topics and tends to be better about warning the implementor about pitfalls to avoid. *—Craig Partridge, BBN*

Letters to the Editor

Executive IETF

Dear Ole:

At the recent IETF in Cambridge, I was amazed to see that more than 600 people were attending. The IETF process has grown almost as fast as the INTEROP Conference. It is becoming increasingly difficult to provide a single forum that is adequate for all people. Perhaps the Internet community could learn a lesson from INTEROP and break the IETF into pieces. Along those lines, let me propose *Executive IETF: The Task Force Within the Task Force.*

Forum for the suits

Executive IETF would be a forum for the suits, a place where issues pertinent to the manager-to-manager communication paradigm could be hammered out. For example, the Executive IETF could establish a Manager-to-Manager Protocol, a specification of the correct procedures for scheduling lunch and exchanging business cards. A Manager-to-Manager MIB could contain the current state of the management team:

```
response    (     CurrentLocation.0 = "At Lunch" ,
                  TechnicalKnowledge.1 = wrong_type
                  FAQ.0 = counter_value_exceeded,
            )
```

Manager management is only one of the topics that could be addressed at the Executive IETF. The Executive IETF could help standardize the procedure for putting on ties, could publish Internet Drafts of fine wines, and could work on informational RFCs containing current buzz words and their pronunciation (a sort of Berlitz guide to the Internet).

If Executive IETF is successful, the model could easily be expanded. SNA IETF and COBOL IETF come to mind as possible outlets for growth.

Sincerely yours,
—Carl Malamud

De facto versus de jure: The arrogance of Standards Organizations

In a recent article emerging from the INTEROP 92 Spring "Great OSI Debate," Richard desJardins starts with the assertion that "OSI is the *de jure* international standardization component of worldwide computer networking." He contrasts OSI with TCP/IP which he characterizes as the "*de facto* standard..." This is flat *wrong,* and continued use of the terms perpetuates a differentiation among standards making bodies arising from organizational arrogance that harms the growth and evolution of the industry.

Legal meaning

It is not clear how this de jure versus de facto stuff got started in discussing telecommunication and information system standards, but the terms have fairly specific meanings in law that are wholly inapplicable to our voluntary systems of standards. "De jure" means legitimate, just or imposed as a matter of law. "De facto" is a contrasting condition characterized as illegitimate, condoned or accepted for practical purposes.

In a world of heterogeneous, voluntary standards making bodies, none of them has a normative right to claim its standards are more legitimate or legally binding than those produced by any others, including individual corporations that have obtained adoption in an open marketplace. Even the CCITT—which is an international body under an intergovernmental organization—does not produce legally binding standards. Indeed, in a highly dynamic, market-driven, virtual communications environment, it is increasingly apparent that the organizations and processes crafted for a slow-paced, hardware-intensive, monopoly provisioning era simply cannot produce acceptable standards.

Grand experiment

The telecommunication and information systems communities over the past twenty years have learned some painful and expensive lessons about what does and does not work. It was in a sense, a grand experiment involving a lot of new technology, innovations, and markets. However, the rhetoric that sustained some standards making processes, organizations, and products in the past has reached the limits of elasticity; and the resulting earthquake of reckoning now underway is loosening some mighty tectonic plates to shear off and subduct—including a hefty seven-layered one.

—Tony Rutkowski, US Sprint

[Rutkowski was formerly the Chief of Telecommunication Regulations and Counselor to the Secretary-General at the ITU in Geneva.]

Comments on "Comments on 'Opinion: OSI Is (Still) a Good Idea'"

Dear Editor:

At the risk of my getting riddled with a few more of Marshall Rose's bullets, please allow me to respond to his comments on my article "OSI Is (Still) a Good Idea," (In this "target rich environment," as he calls it, another couple of targets—surely he'll also want to take aim at the GOSIP Institute's White Paper (elsewhere in this issue, see page 24)—might actually serve to confuse a lesser opponent, but I fear Marshall could handle a few hundred simultaneous targets without a sweat.)

ISO and CCITT have extremely important and strategic international roles to play, which I won't reiterate here, and mostly they play them very well. Their methods are actually very effective for a "meetings/papers" milieu, and they get a *lot* of good work done. I've seen both organizations work extremely effectively. (Go ask Dave Katz (dkatz@cisco.com) about getting IDRP through the ANSI/ISO standardization process.)

Paradigm shift

Nevertheless, in general (and, yes, you heard me say it here): Marshall is right. The truth is that the "problem with ISO" derives fundamentally from the paradigm shift that the Internet community is the beneficiary (care of Bob Kahn, Vint Cerf, and the US Government) and proponent of: (1) open and free electronic discussion and publication of drafts, (2) implementation and interoperability bakeoffs as proof of reality, (3) deployment of infrastructure under the aegis of the same organization. OK, so what's the difference then between Marshall's views and mine? We both agree that ISO is broken, but he doesn't want to fix it whereas I do. Here's how ISO relates to that paradigm shift.

continued on next page

Letters to the Editor *(continued)*

War story

(1) Electronic discussion and publication of drafts? Here's a war story: Back in the mid-1980s (I was working at DARPA at the time), I chaired an ISO task group on Electronic Operation (which meant operating electronically like the Internet did and does). We were able to make some improvements, but here's the real story: the ANSI Secretariat didn't even have a desktop computer, much less e-mail! She had to get one donated by one of the vendors! Many of the rest of us were using e-mail in our organizations, but few companies had access to the Internet. (You had to have a Government contract in those days, as it wasn't until a couple of years later that *MCIMail*—Vint Cerf again!—came along with cheap Internet mail access for the masses.) The point of the story is that the electronic operating infrastructure that the Internet community takes for granted is actually a new paradigm, which the Internet (with plenty of US Government funding) developed and proved, that's true, but which other organizations are now adopting as it becomes widely available, so it's not much of a discriminator for Internet any longer. ANSI and ISO certainly believe in the paradigm: many ANSI and ISO standards are now developed over the Internet. We still have to fix the free distribution of ISO and CCITT standards over the net, but we're getting there. Score this one for Internet and especially for the paradigm: Everybody uses it!

A counter-example

(2) Implementing before standardizing? The IEEE/ANSI/ISO standard on Copper Distributed Data Interface was recently decided after laboratory testing of the two alternatives. Does Internet claim to have developed this method? The original HDLC draft standard was withdrawn in midstream in the 1970s after computer simulations showed up its deficiencies. Notwithstanding these examples, let's give this one to the Internet also. Implementations and proofs of reality should indeed be required before a standard is finalized.

The role of the marketplace

(3) Deployment of infrastructure? ISO depends on its National Bodies such as ANSI to get the infrastructure built. Well, this isn't going to work! You need the Government and the PTTs to build it, like they did for the Internet. In Europe, infrastructure deployers think OSI is spelled "X.25"! That's not going to work either! So what will happen? My view is that in the area of deployment of infrastructure, ISOC and the Internet community have become responsible for worldwide internetworking, just as CCITT and the PTTs are responsible for worldwide telecommunications, and the great unorganized user and vendor community are responsible for worldwide information technology (hardware, languages, media, all of which are ISO, not Internet, standards, or vendor consortia or proprietary standards, and are deployed by users and vendors, not by infrastructure providers or by Internet). The marketplace will choose between Internet, CCITT, ISO, and vendor consortia or proprietary standards, just as it does today. (*PostScript,* XPG, and ONC are doing very well without either ISO or Internet backing. Pascal, **C**, and diskettes are all ISO standards that were implemented before being standardized.) We'll have to score this one for the Internet also, and for the marketplace.

What will become of OSI? In The GOSIP Institute White Paper, it is proposed that OSI and Internet should and will converge. My personal position is that, as a user, I'm comfortable with letting the open systems process continue to take its course. Things are going well for users.

Good OSI technology

There's a lot of good technology available in the Internet. And there's a lot of good technology available in OSI: CLNP, ES–IS, IS–IS, X.400, X.500, Transaction Processing, Remote Database Access. Even the much maligned FTAM will turn some heads with its Phase 3 recovery and restart support, allowing unattended software distribution, back-up over the network, and production bulk data transfer. As proposed in the GOSIP Institute White Paper, let's create the glue to run "anything over anything"—like RFC 1006 (which, I take perverse pleasure in mentioning, has a serious flaw in the event of TCP failure), TUBA, POSIX APIs—and let the market sort it all out. The result will be a vital Internet community by the end of the decade in which the religious wars have been superseded by a productive multiprotocol environment in which everybody interoperates.

Before I get to the punch line, a quick word on Marshall's criticisms of the OSI Skinny Stack. His "set up and tear down" comparison is meaningless, since if you want to compare apples to apples, you have to compare UDP with CLTP, not UDP with ROSE over a full connection-oriented stack (as he seems to imply). For example, NFS has defined NFS over CLTP/CLNP, and it works just as fast as NFS over UDP/IP. On the other hand, if you're going to do a series of exchanges such as a distributed atomic transaction using the new OSI Transaction Processing standard, I claim that setting up a full connection over a custom compiled skinny stack is the proper way to do it, for reliability and efficiency. You can still interoperate with full stacks or with other TP skinny stacks, and there isn't anything else that you'd try to interoperate with!

A particularly fine dinner

Which brings me to my seven "Fearless Predictions." There are plenty of empirical trends and interpretive tea leaves to support my predictions, which I'd be happy to debate in any suitable forum, but where would it get us? Marshall would just dispute the trends and dismiss the tea leaves as nonempirical, so here's an alternative proposal which is undeniably empirical: The desJardins–Rose "$1000 Plus a Particularly Fine Dinner" Challenge. I will donate $1000 to Marshall's favorite charity (I'm tempted to speculate it will be the *OSI Memorial Fund*), plus buy him a "particularly fine dinner," if more than half my predictions do not come true, and I challenge him to put his money where his mouth is as well. We'll let Ole Jacobsen, Carl Malamud and Vint Cerf be the judges (they can develop the scoring system), with Vint as the Master of Ceremonies. In the event of a "near enough tie" (to be defined by the judges), Marshall and I will both pay off. To ensure that the judges are unbiased, they get included in the particularly fine dinner no matter who wins. Members of the press and kibitzers may attend the dinner if they pay their own way and make a $25 contribution to the winning charity. For a ceremony, we'll have the restaurant serve up "crow on a platter," which the loser will publicly partake of after being forced to endure the winner's verbal crowing. Sounds like fun in a perverse sort of way! Marshall, what do you say?

— Dick desJardins

Barely Noticed

Ole,
Regarding the editor's note in the August *ConneXions:* while the phrase "bare witness" has some interesting connotations of its own, I think "bear witness" is the usual phrase.

—Steve Casner

Just so you won't think I'm not paying *any* attention, I'd observe that we *bear* witness and *bare* breasts—idiomatically speaking, of course.
P[hilologically] C[orrect] cheers,

—Mike Padlipsky

Send your letters to:

ConneXions
480 San Antonio Road
Suite 100
Mountain View
CA 94040–1219
USA
Fax: +1 415-949-1779
E-mail:
connexions@interop.com

Announcement and Call for Papers

The 4th *Joint European Networking Conference* (JENC) will be held in Trondheim, Norway, 10–13 May 1993. The conference is organized by RARE (*Réseaux Associés pour la Recherche Européenne*) in cooperation with EARN, IFIP TC6, the Internet Architecture Board, the Internet Society, NORDUnet, and UNINETT.

Goals

The theme of the conference—*European Networking in a Global Context*—acknowledges the fact that networks covering a specific geographical area or networks for a specific user group will only be successful if they are connected to the rest of the world. Global connectivity offered to a large number of users and applications makes networks more interesting to use. Considering that in Europe today only a small fraction of all academic and research staff are network users, it is fair to say that there is still a long road ahead of us. This conference is intended to explore the next steps to take.

The conference addresses all staff from networking service providers of all sizes as well as application developers, policy makers and representatives of funding bodies, advanced user groups and standards organizations. Much emphasis is again placed on cooperation between members of different networking communities, building on the positive experiences of the previous Joint European Networking Conferences. This conference is *the* forum on networking for research in Europe and presents a unique opportunity to meet key people active in the field today.

Venue

Trondheim is the third largest city in Norway. Founded in 997 AD by the Viking King Olav Tryggvason, the city lies on a bend in the river on the edge of Trondheim fjord. During the Middle Ages, Trondheim was the capital of Norway and a major place of pilgrimage. Today Trondheim is a national centre of technical education, research and development. The campus of the Norwegian Institute of Technology (NTH), part of the University of Trondheim, offers good conference facilities and provides an excellent background for a leading network conference such as the JENC. Trondheim is easily reached. Trondheim Airport has direct flights to Oslo, Copenhagen and Stockholm and further connections to European and overseas destinations.

Papers

As at previous Joint Networking Conferences, the programme will include a combination of solicited and submitted papers. Slots of 30 minutes will be assigned to speakers, which include time for questions and discussion. In special cases, two slots of 30 minutes may be assigned to one speaker. Two-page summaries of proposed papers should arrive at the programme chair (Bernhard Plattner) not later than November 1, mentioning the topic against which the author would like his paper assessed. Accepted papers must be submitted in full to the programme chair not later than April 10, 1993, in order to provide a full set of papers to the Conference participants. The full papers will be evaluated in a formal review process; good papers will be selected for publication in a special issue of *Computer Networks and ISDN Systems*.

Topics

Submitted papers should be related to one of the major topics of the general outline of the conference given below:

Lower Layer Technology:
- Transmission Technology
- Protocols
- Operational Experience
- Pilot Projects

Mail And Messaging:
- Service Requirements and Performance
- End-user Requirements
- Mail Routing

Network Infrastructure:
- Network Management
- Quality of Service Definition and Measurement
- Global Interconnection Issues
- Reaching "Off-campus" and Mobile Users

Network Applications:
- Directory Services
- Multimedia Mail and Document Transfer
- Management of Distributed Applications
- Group Communications and Conferencing
- Information Services
- Distance Learning

Users:
- User Requirements
- The Network as Information Resource
- Education and Training for Users
- Cataloging and Discovering Network Resources
- Discipline-oriented User Groups

Security:
- Administrative Requirements and Procedures
- National and International Policy and Legal Issues
- Authentication and Authorization
- Secure Applications

Policy, Funding, and Strategies:
- Economic Impact of Networking
- Electronic Publishing and Intellectual Property Rights
- International Export versus Legal Restrictions

Status Reports of National Initiatives and European Projects:
- Results of COSINE
- National Networks and Activities
- Central and Eastern European Activities
- Standards
- EARN/EurOpen/RIPE
- EMPB/EBONE

Posters and demonstrations

As in previous years, a poster wall will be available for the display of posters. Participants are invited to submit a poster presentation of the project they work on or a topic of common interest to the conference participants. The programme committee will select the best two posters during the conference for inclusion in the conference proceedings. The best poster wins a free registration to the next JENC.

During the conference there will again be the opportunity for participants to present their project or activities in the form of a demonstration, either as part of their presentation or separately. Requests for demonstrations should be made through the RARE Secretariat, specifying technical requirements before November 1 (later proposals will be taken into account but can not be guaranteed booth space). X.25 and IP connectivity will be provided. The programme committee will select the best demonstrator who will win a free registration to the next JENC.

continued on next page

Announcement/Call for Papers *(continued)*

Tutorials

This year for the first time tutorials will be organized. They will be held at the end of the conference, on Thursday afternoon and Friday morning. Proposals for half-day or full-day tutorials should arrive at the programme chair before the 1st of November, 1992. The programme committee also invites prospective attendees to send a list of topics on which a Tutorial would be appreciated to the programme chair before November 1, 1992.

Conference format

Based on the successful conferences of the previous years, the format of the Conference has not changed, with the exception of the addition of Tutorials. The conference starts at 14:00 on Monday 10 May 1993 and runs until 12:30 on Thursday 13 May 1993. No sessions will be scheduled for the Wednesday morning, allowing participants to organize BOFs. As usual many international meetings will be scheduled around the Conference.

Important dates

2-page summaries of proposed papers due:	November 1, 1992
Deadline for proposed tutorials:	November 1, 1992
Deadline for proposed demonstrations:	December 15, 1992
Notification of acceptance:	End January, 1993
Full papers & final demo descriptions due:	April 10, 1993

Further information

RARE Secretariat
Singel 466-468
NL-1017 AW Amsterdam
The Netherlands
Tel: +31 20 639 1131
Fax: +31 20 639 3289
E-mail: raresec@rare.nl

USENIX Online Library/Index

What is it?

The USENIX online index is an electronically available list of papers published by the USENIX Association and related groups. The index is kept as a simple ASCII file, in refer/bib format, sorted by author, and contains information about papers published in USENIX conference and workshop proceedings, newsletters, journal, and the like. The index is updated approximately monthly.

How to get the index

The index is available online from UUNET, either via a mail server or anonymous FTP. The index is about 200K bytes, and available only in its entirety. To get it as mail, send mail to `library@uunet.uu.net` with "send bibliography" as the contents of your message. To get it via FTP from `ftp.uu.net`, login as "anonymous" with your e-mail address as the password. Then:

```
ftp> cd library
250 CWD command successful.
ftp> get bibliography
```

The help file can be retrieved with "send help" or as the FTP file `help.bibliography`. (There is no person associated with the library address and it will never be read by human eyes.)

How to access information

To build the indices so you can easily access information, run *indxbib* on the bibliography: `indxbib file.name`. You can then pull information from the file by running *lookbib*. You can either build *refer* files or run *lookbib* interactively. For example, the following command would put all entries which refer to "Smith" into a file called "stuff":

```
echo smith | lookbib bibliography > stuff
```

Or you could interact with the index by saying:

```
lookbib bibliography
```

It will ask you if you want instructions when it starts, answer yes. Then at the prompt, for example:

```
>smith
```

will list references to smith (upper/lower case doesn't matter).

As of this date, we have not yet set up the online papers. When this capability is provided, we will announce it on the net and this document will be updated with retrieval information.

USENIX publications indexed

- Conference proceedings, Workshop proceedings
- Computing Systems Journal, Newsletters
- EurOpen (formerly EUUG—European Unix Users Group):
- Conference proceedings, Newsletters (1982–1989)

(Other sources are being continually evaluated and will be included as deemed suitable).

Fields used in the index

The standard *bib*/*refer* formats are used. These include:

%A	Author (may be multiple entries)
%T	Title of article
%P	Page number(s)
%W	Primary author's institution
%I	Issuer/publisher
%B	Conference proceedings or book title
%J	Name of newsletter or journal
%D	Date of publication or conference
%C	Location of conference
%V	Volume number
%N	Number within volume
%O	Other comments (e.g., "Abstract only")

These fields may be extended to include other information such as identifier for retrieval, keywords, online format of paper (*PostScript, troff,* etc.), language (if other than English), etc.

More information

For additional information about the online index and library, and/or instructions for donating papers, contact:

USENIX Association
2560 Ninth Street
Suite 215
Berkeley CA 94710
USA
`index@usenix.org`

Call for Papers

ACM SIGCOMM '93—Communications Architectures, Protocols and Applications will be held September 15–17th, 1993 (Tutorials September 13–14) in San Francisco, California, USA. The conference provides an international forum for the presentation and discussion of communication network applications and technologies, as well as recent advances and proposals on communication architectures, protocols, algorithms, and applications.

Topics

Authors are invited to submit full papers concerned with both theory and practice. The areas of interest for the conference include, but are not limited to the following:

- Analysis/design of computer network architectures and algorithms
- Innovative results in local area networks
- Mixed-media networks, high-speed networks
- Routing and addressing, support for mobile hosts
- Resource sharing in distributed systems
- Network management
- Distributed operating systems and databases
- Protocol specification, verification, and analysis

The conference is single-track and the number of sessions and papers varies with the number of accepted submissions. In the past the conference has accepted about 30 papers from well over 100 submissions. Successful submissions, both practically and theoretically oriented, typically report results firmly substantiated by experiment, implementation, simulation, or mathematical analysis.

Submissions

All submissions should be accompanied by a cover letter containing a list of all the authors, their affiliation, telephone numbers, electronic mail addresses, and facsimile telephone numbers. Papers must be less than 20 double-spaced pages long and should have an abstract of 100-150 words. All submitted papers will be reviewed and will be judged with respect to their quality and relevance.

Authors considering a submission are encouraged to obtain more information about the topical areas and characteristics of past selected papers by sending an electronic mail message to `sigcomm93-author-info@umbc3.umbc.edu`. Authors of accepted papers will be expected to sign an ACM copyright release form. The Proceedings will be distributed at the conference and published as a special issue of *ACM SIGCOMM Computer Communication Review*. The program committee will also select a few papers for possible publication in the IEEE/ACM Transactions on Networking.

Submit five paper copies or e-mail one *PostScript* copy of each paper to the program chairman:

Deepinder Sidhu
Department of Computer Science
University of Maryland–BC & UMIACS
Baltimore, MD 21228, USA
E-mail: `dsidhu@umbc3.umbc.edu`
Phone: +1 410-455-3028 • Fax: +1 410-455-3969

Note: Any information about the authors must appear on a separate cover page, so that it can be removed before papers are sent to reviewers.

For more information about the conference (as opposed to paper submissions), e-mail to: sigcomm93@cse.ucsc.edu.

Tutorial proposals should be submitted to the Tutorial Chair:

Ian F. Akyildiz
School of Electrical Engineering
Georgia Tech
Atlanta, GA 30332, USA
ian@cc.gatech.edu
Phone: +1 404-894-5141
Fax: +1 404-894-3188

Student Paper Award

Papers submitted by students will enter a student-paper award contest. Among the accepted papers, a maximum of four outstanding papers will be awarded (1) full conference registration and (2) a travel grant of $500 US dollars. To be eligible, the student must be the sole author, or the first author and primary contributor to the paper. A cover letter must identify the paper as a candidate for this competition.

Important dates

Deadline for paper submissions/tutorial proposals: 22 February 1993

Notification of acceptance: 7 May 1993

Camera ready papers due: 11 June 1993

OSI "Skinny Stack" software available

Source of the X/osi Skinny Stack software is now available by anonymous FTP from pluto.ulcc.ac.uk(192.12.72.4). The top-level README file should be read. The software is in directory ulcc/Xosi. All of it can be picked up at once in the compressed *tar* file xosidist.tar.Z, or the directory dist may be examined. README files at each level document the material.

Improvements and additions (especially to the documentation) are likely at irregular and frequent intervals.

—*Peter Furniss* (P.Furniss@ulcc.ac.uk)

Trusted Systems Interoperability Group

The *Trusted Systems Interoperability Group* (TSIG) is a consortium of vendors, systems integrators and government types who are interested in addressing issues related to the interoperation of multilevel secure systems and compartmented mode workstations. TSIG is an open forum that meets about every 10–12 weeks. Meeting announcements are posted to the general mailing list (see below).

Mailing list

To get on the general mailing list, send an e-mail message to: tsig-request@wdl1.wdl.loral.com. The five TSIG working groups are *Commercial IP Security Option* (CIPSO), *Trusted Network File System* (TNFS), *Trusted Session* (TSESS), *Trusted X-windows* and *Trusted Administration* (TADMIN). Each has their own mailing lists at wdl1.wdl.loral.com. CIPSO and TNFS are joint IETF/TSIG working groups and a subset of TADMIN is the proposed IETF Audit Information Transfer Protocol WG.

CONNEXIONS
480 San Antonio Road
Suite 100
Mountain View, CA 94040
415-941-3399
FAX: 415-949-1779

ADDRESS CORRECTION
REQUESTED

FIRST CLASS MAIL
U.S. POSTAGE
PAID
SAN JOSE, CA
PERMIT NO. 1